IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2019, accepted July 25, 2019, date of publication August 1, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932412

Visualization of the Image Geometric Transformation Group Based on Riemannian Manifold

TIANCI LIU[1,2,3,4,5], ZELIN SHI[1,2,4,5], AND YUNPENG LIU[1,2,4,5]

[1]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang 110016, China
[5]The Key Laboratory of Image Understanding and Computer Vision, Shenyang 110016, China

Corresponding authors: Tianci Liu (liutianci@sia.cn) and Zelin Shi (zlshi@sia.cn)

This work was supported by the Innovation Fund of Chinese Academy of Sciences under Grant Y8K4160401.

ABSTRACT Geometric transformations of images are the predominant factor, which influences the effectiveness of visual tracking and detection tasks in computer vision. Naturally, although it makes significant sense to grasp the process of image geometric transformations, the numerical relationship of geometric transformations cannot be revealed directly from images themselves. Even if the geometric transformation matrices form the three-dimensional special linear group, $\mathrm{Sl}(3,\mathbb{R})$ group, it is difficult to comprehend the manifold of this invisible visual motion, which resides in the high-dimensional space. Furthermore, the main challenge is the deficiency of analytic expressions of the Riemannian logarithmic map to compute the geodesic distance on the $\mathrm{Sl}(3,\mathbb{R})$ manifold. Facing these issues, this paper comes up with a novel approach to visualize the geometric transformation in images by presenting a new metric, and then, computes a set of coordinate-vectors in the three-dimensional state transition space for visualization using the Riemannian stress majorization. The superiority of the presented framework for visualization, in terms of accuracy and efficiency, is demonstrated through abundant experiments on aerial images and moving objects.

INDEX TERMS Geometric transformation, visualization, motion group, Riemannian manifold.

I. INTRODUCTION

In computer vision, the characteristics of visual patterns can be described by some transformations whose state transitions are regulated by action groups. Taking the geometry of images into account, these action groups deriving from visual data are always relevant to some particular matrix manifolds [1] through the invariance correspondingly constrained on them. Most of these transformations have the matrix structure of Lie group, which gives rise to the representation of matrix manifolds. Specifically, the image geometric transformations have gained extensive attention in various applications including object detection, visual tracking and image registration [2]–[5]. Although the geometric transformations of images are the predominant factor which influences the effectiveness of visual tracking and detection tasks, this crucial transformation may not be measured numerically from images directly. If the changing process of this motion can be visualized from the abstract mathematical space, it will provide more possibilities to obtain better effectiveness of various visual tasks. Unfortunately, the relevant research of the visualization of the image geometric transformations is still in the blank stage, which becomes the obstacle to comprehend this motion group.

From the mathematical perspective, the geometric transformations of visual data form the structure of Lie group, $\mathrm{Sl}(3,\mathbb{R})$, which leads to the representation of the matrix manifold. However, on the one hand, the logarithm of a real invertible matrix may not exist generally, and if it exists it may not be unique. Furthermore, as the exponential map isn't onto and one-to-one in general, its inverse map can be defined only in a certain neighborhood of the identity element. In this case, the matrix logarithm map may not be the best choice to measure the geodesic length on the $\mathrm{Sl}(3,\mathbb{R})$ manifold if the samples are beyond the neighborhood of the identity element. Therefore, the distance defined by the simple matrix

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

logarithmic operation may not measure the true geodesic distance on the manifold so that the visualization purpose is out of work. Based on these reasons, in this paper, we focus on the geometric properties of the $Sl(3,\mathbb{R})$ manifold to measure the geodesic length instead of just using the simple matrix exponential and logarithm maps.

On the other hand, Riemannian manifolds are topologically abstract mathematical definitions such that few relevant concepts of manifolds exist in the visualization aspect. Even if manifolds take advantages of the geometry of visual data, realizing the changing process happened in the non-Euclidean space is difficult. Considering this fact, we will benefit a lot if the abstract manifolds can be visualized to cause more intuitive results. Moreover, although we want to take advantage of the Riemannian structure of the $Sl(3,\mathbb{R})$ manifold to solve the limitation of the algebraic exponential and logarithm maps, the inexistence of the analytic expression of the Riemannian logarithmic map on the $Sl(3,\mathbb{R})$ should be considered.

This article is an extended version of our initial work in the conference paper [6]. Here, we extend our work not only by providing more mathematical backgrounds, but also illustrating the details of Riemannian mean method. Last but not least, we significantly define three objective evaluation indicators and conduct more extensive experiments to provide a systematic analysis of different methods. In summary, the main innovations and extensions of this article based on our conference paper [6] are briefly concluded as follows,

- We define three objective evaluation indicators to numerically measure the quality of visualization results.
- We add four state-of-the-art approaches of visualization for comparison and report the relevant results in Table.1-Table.3.
- We add extensive experiments to provide a systematic analysis of our method with other compared algorithms.
- We provide the performance of the proposed method with the increased number of experimental image sequences.

The main contributions of this paper are listed as follows,

- We first explore the possibility of proposing a Riemannian geometry-based framework to visualize the $Sl(3,\mathbb{R})$ group from the image geometric transformation.
- For the essential metrics used in our method, comparing with the conventional metric deriving from the simple algebraic logarithmic map, we present a new method to approximate the geodesic distance on the $Sl(3,\mathbb{R})$ manifold by derivating the analytic expression of the corresponding Riemannian exponential map.
- For the quantitative evaluation of the proposed method, we particularly define three objective evaluation indicators to measure the quality of the visualization performance.

The remainder of this paper is organized as follows. In Section II, we provide the description of the related works. In Section III, we briefly introduce the mathematical notions of Riemannian manifolds and Lie groups. In Section IV, we describe the details of the proposed visualization framework for $Sl(3,\mathbb{R})$ and its implementation. In Section V, the experiments on the synthetic and real images demonstrate the effectiveness of our proposed approach. Section VII makes the conclusion of the whole paper.

II. RELATED WORKS

Nowadays, the deep learning methods have been developed as the most popular techniques in computer vision, which achieves many breakthroughs in the area of image classification [7]–[9] and object detection [10]–[12]. In view of these successful experience, some researchers extended deep learning algorithms into the visualization area to play an important role. Firstly, in 2013, Simonyan *et al.* [13] visualized image classification models and their corresponding class saliency maps in the deep convolutional networks. Subsequently, in 2014, Zeiler and Fergus [14] proposed a novel visualization framework to understand the image classification models of the convolutional neural network (CNN). They visualized the features of each intermediate convolutional layers using deconvolutional networks, which aims to find interpretations and guidance for deep models. Recently, Bojarski *et al.* [15] represented a new approach, VisualBackProp, to visualize the most significant pixels of the input image for CNN-based classification. They also provided the rigorous mathematical analysis as the theoretical demonstration of the visualization model. Moreover, deep feature representations learned from CNNs were used to transform the image styles in [16], [17]. In a word, these literature mainly focus on the visualization of the feature representations or gradient maps to interpret the CNN model instead of the Riemannian manifolds.

Besides deep learning methods, in recent years, the topic of geometric dimensionality reduction algorithms for Riemannian manifold has attracted more and more attention. As for the research of dimensionality reduction from manifold to manifold, the development of this topic can be summarized as follows. Harandi first represented the manifold dimensionality reduction method on the Symmetric Positive Definite (SPD) matrices in [18], which can be regarded as the beginning work about this topic. Based on this conference paper, Harandi extended his initial work to the journal paper [19] by introducing an unsupervised dimensionality reduction approach and deriving variants of corresponding unsupervised/supervised dimensionality reduction algorithms based on the Jeffrey divergence. And then, Huang also makes a great contribution to the manifold dimensionality reduction field. Huang proposed the Log-Euclidean Metric Learning (LEML) on the SPD manifold in [20]. LEML transforms the dimensionality reduction from the SPD manifold to the corresponding tangent space, which aims to learn a tangent map that can directly transform the matrix logarithms from the original tangent space to a new tangent space of more discriminability by the use of Log-Euclidean metric. For the Projection Metric Learning (PML) in [21], Huang

embedded the points on the Grassmannian into the points on the SPD manifold under the projection framework by embedding Grassmann manifolds onto the space of symmetric matrices, which is actually a special type of dimensionality reduction for the SPD manifold by the use of the projection metric on the Grassmann manifold. Next, Liu proposed a generalized Grassmannian framework of dimensionality reduction for the Grassmann manifold in [22]. His framework is suitable for any metric on the Grassmann manifold rather than being limited to the projection framework as PML. In his model, he directly proposed a geometry-aware dimensionality reduction for the Grassmann manifold to obtain a lower-dimensional manifold where better classification can be achieved. Because his model does not depend on any additional intermediary, in theory, any distance metric can be used directly. However, these manifold dimensionality reduction methods above are specially designed for SPD manifolds and the Grassmann manifolds for better classification results. To the best of our knowledge, there does not exist a manifold dimensionality reduction method specially designed for the $\mathrm{Sl}(3,\mathbb{R})$ manifold, which limits us to choose this kind of approach for the visualization purpose.

On the other hand, several studies that investigated the visual motion of the image geometric transformation have attracted increasing attention. At first, E. Begelfor and M. Werman in [23] presented a novel group-invariant Lie-normal distributions on the $\mathrm{Sl}(3,\mathbb{R})$ group to enhance the accuracy of object recognition. Moreover, the parameters of the probabilities are estimated by a simple algorithm, which leads to the easy computation of the density. Then, S. Benhimane and E. Malis proposed a new homography-based approach to visual tracking and servoing through the optimization of the second-order minimization in [24]. They presented a real-time framework without computing the Hessian matrix to improve the tracking speed and reduce the computational complexity. In [25], S. Lee and M. Choi proposed a geometric coordinate-invariant approach designed for the image registration by taking the geometry of $\mathrm{Sl}(3,\mathbb{R})$ groups into account. They generalized the Nelder-Mead optimization to the $\mathrm{Sl}(3,\mathbb{R})$ group and other Lie groups, which makes the geometric direct search algorithms lead to superior performances by taking advantages of the geometric properties. However, all these literature just simply consider the algebraic exponential and logarithmic maps as the approximation to the Riemannian exponential and logarithmic maps respectively. In this case, the matrix logarithm map may not be the best choice to measure the geodesic length on the $\mathrm{Sl}(3,\mathbb{R})$ manifold if the samples are beyond the neighborhood of the identity element.

To the best of our knowledge, this paper is the first attempt to propose the approach which concerns about the geodesic distance on $\mathrm{Sl}(3,\mathbb{R})$ manifold with the Riemannian metric instead of the simple algebraic exponential and logarithmic maps for visualization. In response to this limitation, we propose a new framework which not only present a new metric to measure the true geodesic distance on the manifold but also visualize the process of the motion groups.

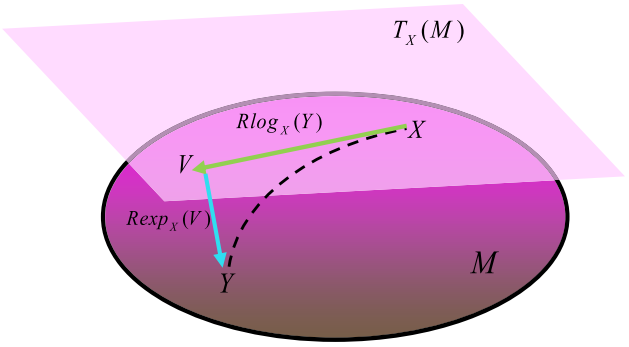


FIGURE 1. Conceptual illustrations of Riemannian exponential and Riemannian logarithmic mapping.

III. RIEMANNIAN MANIFOLD AND LIE GROUP REPRESENTATION

A. RIEMANNIAN MANIFOLD

From the geometric perspective, the manifold is the topological space that is locally homeomorphic to a Euclidean space. In other words, every point on the manifold is locally similar as a Euclidean space. Given a concrete example, the earth can be regraded as a manifold where each point is approximatively treated as a Euclidean space. A Riemannian manifold is a manifold which is equipped with the Riemannian metric, a symmetric and positive-definite inner product. Due to the space constraint, for the further related mathematical details, readers can refer to the literature [26]–[28]. From now on, we use manifolds to stand for Riemannian manifolds in the rest of this article.

Mathematically, every point X on the manifold M has a tangent space at X, denoted by $\mathrm{T_XM}$. Given any two points $\mathrm{X, Y} \in \mathrm{M}$, the shortest path between them is the geodesic curve on the manifold, which is denoted as $\gamma(\mathrm{t})$ with $t \in [0, 1]$. Comprehensibly, the tangent space $\mathrm{T_XM}$ can be regarded as the vector space including all tangent vectors that correspond to all geodesics connecting to X. The tangent space $\mathrm{T_XM}$ consists of tangent vectors $\mathrm{V} \in \mathrm{T_XM}$ which corresponds to the geodesic between X and Y on the manifold. The Riemannian logarithmic map operator $\mathrm{Rlog_X : M \to T_XM}$ achieves the projection $\mathrm{Rlog_X(Y) = V}$, which projects the geodesic (i.e., the dashed line in Fig.1) between X and Y from the original space of matrix manifold M to the tangent space $\mathrm{T_XM}$. Correspondingly, the Riemannian exponential map operator $\mathrm{Rexp_X : T_XM \to M}$ realizes the projection $\mathrm{Rexp_X(V) = Y}$, which makes the tangent vector V go back into the manifold. Through this approach, the length of geodesics is equivalent to that of the corresponding tangent vector obtained by the Riemannian logarithmic map. Conceptual illustrations of these operations are shown in Fig.1.

B. LIE GROUP REPRESENTATION OF GEOMETRIC TRANSFORMATION

In computer vision, the geometric transformation is represented by a linear equation in the homogeneous coordinates

with a nonsingular 3×3 matrix, which is called **homography**. More specifically, let $I_1(p)$ denote the gray level of an arbitrary image I_1. The gray level of the transformed image I_2 which goes through the geometric transformation from I_1 can be represented as

$$I_2(T(p;r)) = I_1(p) \tag{1}$$

where $p = (x, y)^T$ represents the position of pixels and $r = (r_1, r_2, \cdots, r_8)^T$ is the parameter vector of the geometric transformation matrix.

In this case, the relationship of pixels stemming from the geometric transformation can be written as

$$T(p;r) = \frac{1}{r_7 + r_8 y + 1}\begin{bmatrix} r_1 x + r_2 y + r_5 \\ r_3 x + r_4 y + r_6 \end{bmatrix} \tag{2}$$

Here, Eq.2 represents the geometric transformation from the original image I_1 to the transformed image I_2.

Consequently, the homography can be defined correspondingly by

$$H(r) = \begin{bmatrix} r_1 & r_2 & r_5 \\ r_3 & r_4 & r_6 \\ r_7 & r_8 & 1 \end{bmatrix} \tag{3}$$

Note that matrices differing only by scalar multiplication represent the same homography referred to [23]. For the sake of removing the scale ambiguity of homogeneous coordinates, we normalize Eq.3 with the unit determinant. Mathematically, referred to [2], homographies operated in this approach own the manifold geometry with a special linear group, which means the normalized homographies can be identified with the three-dimensional special linear group $\mathrm{Sl}(3,\mathbb{R})$:

$$\mathrm{Sl}(3,\mathbb{R}) = \{\mathrm{X} \in \mathrm{Gl}(3,\mathbb{R}) : \det(\mathrm{X}) = 1\} \tag{4}$$

where $\mathrm{Gl}(3,\mathbb{R})$ is the general linear group consisting of 3×3 invertible real matrices. As for the Lie group, it is known that the matrix Lie group is endowed with a differentiable structure of Riemannian manifold. Each element in the Lie group corresponds to a tangent space where local neighborhoods of that element may be expressed. More specifically, the tangent space of the identity point generates the Lie algebra. The Lie algebra associated to $\mathrm{Sl}(3,\mathbb{R})$ group is $\mathrm{sl}(3,\mathbb{R})$, which represents the set of 3×3 real matrices with null trace. For more notions of Lie groups and Lie algebras, readers can refer to [29], [30].

IV. THE PROPOSED METHOD

In this section, the content is organized as follows. Firstly, we first derive the geodesic expression on the $\mathrm{Sl}(3,\mathbb{R})$ manifold and then give the formulation of its Riemannian exponential map in IV-A. Secondly, we describe the approach to compute the mean point on the $\mathrm{Sl}(3,\mathbb{R})$ manifold in IV-B. Last but not least, we present a novel approach to visualize the geometric transformation of images in IV-C.

A. GEODESIC DISTANCE OF GEOMETRIC TRANSFORMATION GROUP

Let $X_1, X_2 \in \mathrm{Sl}(3,\mathbb{R})$ be any two points on the $\mathrm{Sl}(3,\mathbb{R})$ group. The shortest curve between X_1 and X_2 is the geodesic denoted by $\gamma(t)$ with $t \in [0, 1]$, and then $v(t) \in \mathrm{sl}(3,\mathbb{R})$ denotes the corresponding tangent vector. For any two tangent vectors $A, B \in \mathrm{sl}(3,\mathbb{R})$, the inner product in the tangent space is defined as

$$\langle A, B\rangle = Tr(A^T B) \tag{5}$$

The function $\exp(\cdot)$ represents the ordinary matrix exponential, which is expressed as

$$\exp(x) = \sum_{m=0}^{\infty} \frac{x^m}{m!} \tag{6}$$

And the symbol $\|\cdot\|$ denotes the matrix Frobenius norm.

Remark 1: If $\gamma(t)$ is the geodesic on the $\mathrm{Sl}(3,\mathbb{R})$ group, it satisfies that

$$\frac{d\gamma(t)}{dt} = \gamma(t)v(t), \quad \text{with} \quad \gamma(0) = X_1, \gamma(1) = X_2 \tag{7}$$

and the tangent vector $v(t)$ which makes $\int_0^1 \langle v(t), v(t)\rangle\, dt$ smallest satisfies that

$$\frac{dv(t)}{dt} = [v^T(t), v(t)]^1 \tag{8}$$

Proof: Here, we demonstrate this problem by the calculus of variations [31].

Given $\eta(t) \in \mathrm{sl}(3,\mathbb{R})$, the perturbation ε of $\gamma(t)$ in the direction of $\eta(t)$ is $\exp(\varepsilon\eta(t))$, and then we can obtain

$$\gamma(t,\varepsilon) = \gamma(t)\exp(\varepsilon\eta(t)) = \gamma(t)(id + (\varepsilon\eta(t))) + o(\varepsilon) \tag{9}$$

where the boundary conditions satisfy that

$$\gamma(0,\varepsilon) = X_1, \gamma(1,\varepsilon) = X_2, \quad \text{with} \quad \eta(0) = \eta(1) = 0 \tag{10}$$

Similarly, the perturbation ε acts on $v(t)$ is given by

$$v(t,\varepsilon) = v(t) + \varepsilon\frac{dv(t)}{dt} + o(\varepsilon) \tag{11}$$

On the one hand, combining with Eq.7 and Eq.9, we have

$$\begin{aligned} \frac{d\gamma(t,\varepsilon)}{dt} &= \frac{d(\gamma(t) + \varepsilon\gamma(t)\eta(t) + o(\varepsilon))}{dt} \\ &= \frac{d\gamma(t)}{dt} + \varepsilon\gamma(t)\frac{d\eta(t)}{dt} + \varepsilon\frac{d\gamma(t)}{dt}\eta(t) + o(\varepsilon) \\ &= \gamma(t)v(t) + \varepsilon\gamma(t)\frac{d\eta(t)}{dt} + \varepsilon\gamma(t)v(t)\eta(t) + o(\varepsilon) \end{aligned} \tag{12}$$

On the other hand, according to Eq.9 and Eq.11, we have

$$\begin{aligned} \frac{d\gamma(t,\varepsilon)}{dt} &= \gamma(t,\varepsilon)v(t,\varepsilon) \\ &= (\gamma(t) + \varepsilon\gamma(t)\eta(t))\left(v(t) + \varepsilon\frac{dv(t)}{dt}\right) + o(\varepsilon) \\ &= \gamma(t)v(t) + \varepsilon\gamma(t)\frac{dv(t)}{dt} + \varepsilon\gamma(t)\eta(t)v(t) + o(\varepsilon) \end{aligned} \tag{13}$$

[1] Here, for two arbitrary matrices A and B, we define the operator $[\cdot]$ as $[A^T, B] = A^T B - BA^T$

Comparing with Eq.12 and Eq.13, we find

$$\frac{dv(t)}{dt} = \frac{d\eta(t)}{dt} - [\eta(t), v(t)] \tag{14}$$

Furthermore, combining with Eq.14, the geodesic $\gamma(t)$ should have the property that

$$\begin{aligned}
&\frac{d}{d\varepsilon}\int_0^1 \left\langle v(t) + \varepsilon\frac{dv(t)}{dt}, v(t) + \varepsilon\frac{dv(t)}{dt}\right\rangle dt\Bigg|_{\varepsilon=0} \\
&= 2\int_0^1 \left\langle v(t), \frac{dv(t)}{dt}\right\rangle dt\Bigg|_{\varepsilon=0} \\
&= 2\int_0^1 \left\langle v(t), \frac{d\eta(t)}{dt} - [\eta(t), v(t)]\right\rangle dt \\
&= 2\int_0^1 \left\langle v(t), \frac{d\eta(t)}{dt}\right\rangle dt + 2\int_0^1 \left\langle \left[v^T(t), v(t)\right], \eta(t)\right\rangle dt \\
&= 2\int_0^1 \left\langle -\frac{dv(t)}{dt} + \left[v^T(t), v(t)\right], \eta(t)\right\rangle dt \equiv 0
\end{aligned} \tag{15}$$

Note that the second term of the formulation in the next-to-last row and the first term in the last row of Eq.15 are obtained from the integration by parts and the fact $\eta(0) = \eta(1) = 0$.

As $\eta(t)$ is the arbitrary variation, according to Eq.15, we have

$$\frac{dv(t)}{dt} = \left[v^T(t), v(t)\right] \tag{16}$$

□

Combing with Eq.7 and Eq.16, we can get the formula of the geodesic on $\mathrm{Sl}(3, \mathbb{R})$ group referred to [32]

$$\gamma(t) = \exp\left(-tV_0{}^T\right)\exp\left(t\left(V_0 + V_0{}^T\right)\right)X_0 \tag{17}$$

where $\gamma(0) = X_0$, $V(0) = V_0$ with $V(t) \in \mathrm{sl}(3, \mathbb{R})$.

Above all, we can naturally get the analytic expression of the Riemannian exponential map on the $\mathrm{Sl}(3, \mathbb{R})$ group

$$\mathrm{Rexp}_e(X) = \mathrm{Rexp}(X) = \exp(-X^T)\exp(X + X^T) \tag{18}$$

However, Eq.18 is only the analytic expression which measures the geodesic starting from the identity in the direction X of length $\|X\|$. It is generalized known that there does not exist a closed-form formulation to measure the geodesic length between I and points on the $\mathrm{Sl}(3, \mathbb{R})$ manifold due to the inexistence of the analytic expression of Riemannian logarithmic map. By virtue of Eq.18, the solution to this issue is to calculate the geodesic arc length numerically by the nonlinear least squared method [33] based on the Riemannian exponential map.

B. MEAN ON THE RIEMANNIAN MANIFOLD

Now in this part, we need to compute the mean of data points for the tangent space of the anchor point. Given the set of points $\{X_i\}_{i=1,2,\cdots,n}$ on a Riemannian manifold M, the Karcher mean [34] of them represented by u can be written as

$$u = \arg\min_{X\in M} \sum_{i=1}^{n} d^2(X_i, X) \tag{19}$$

where $d(\cdot)$ represents the geodesic distance on the manifold and n is the number of data samples.

Considering the definition of Eq.19, the objective function of our model which measures the Riemannian squared distance is defined by

$$L(X) = \sum_{i=1}^{n} d^2(X_i, X) \tag{20}$$

Remark 2: The gradient of the Riemannian squared distance function can be derived as

$$\nabla f(X) = \nabla_X d^2(X, Y) = -2\mathrm{Rlog}_X(Y) \tag{21}$$

Proof: This property is well known in the literature [35], [36]. For the details, please refer to the Theorem.1 in [37]. □

Differentiating Eq.20 with respect to X, according to Remark.2, we can obtain that

$$\nabla_X\left(\sum_{i=1}^{n} d^2(X_i, X)\right) = -2\sum_{i=1}^{n}\mathrm{Rlog}_X(X_i) \tag{22}$$

Thus, u is the solution of X to Eq.23

$$\sum_{i=1}^{n}\mathrm{Rlog}_X(X_i) = 0 \tag{23}$$

The solution of Eq.23 can be obtained by iterative optimization through the gradient descent method [38] on the Riemannian manifolds:

$$u^{t+1} = \mathrm{Rexp}_{u^t}\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{Rlog}_{u^t}(X_i)\right) \tag{24}$$

Nevertheless, the method mentioned above mainly calculates the first-order approximations to u. Note that if X_i is far away from the identity element, the error of this approximation may be undesirable. Taking account of this issue, we make the procedures as follows. Firstly, we left-multiply every point by the inverse of u to compute the residual error ΔX_i of each point according to Eq.26, and then project the total residues of the original samples Δu back into the tangent space. The computation formula of Δu is written as

$$\Delta u = \mathrm{Rexp}\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{Rlog}(\Delta X_i)\right) \tag{25}$$

where

$$\Delta X_i = u^{-1}X_i \tag{26}$$

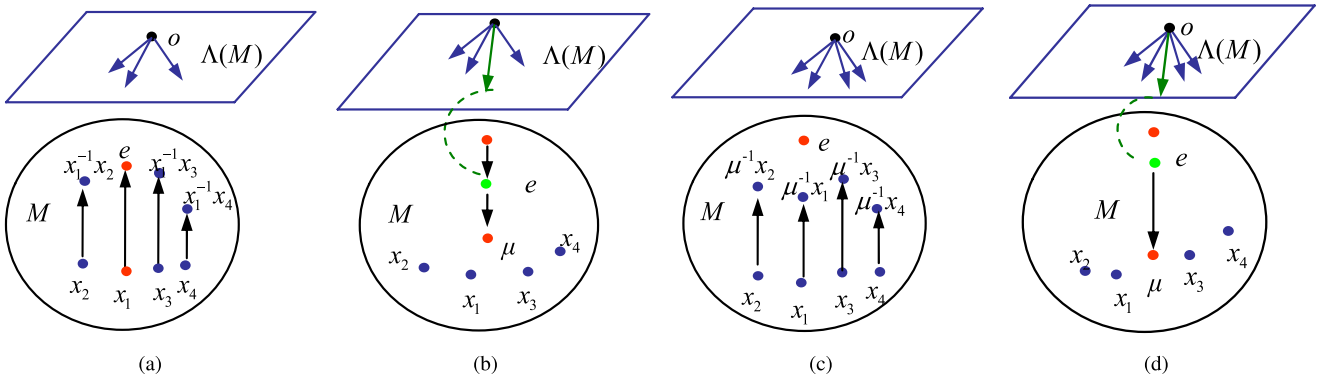


FIGURE 2. Conceptual illustrations of Riemannian mean algorithm on Sl(3, $\mathbb{R}$): (a) Remap each sample point by left-multiplying the inverse of initial u (i.e.,X_1), and then map them to the tangent space of u (i.e., $\Lambda(M)$); (b) Compute the total residues of the original samples Δu and project it back into the manifold M by Eq.25, and then update the mean point; (c)Remap all sample points by the newly updated mean, and then project them into $\Lambda(M)$; (d) Repeat the procedure (b) until convergence.

The algorithm is described in the following and illustrated by Fig.2. In practice, the value of the convergence parameter ϵ is set to 10^{-4}.

Algorithm 1 Riemannian Mean Method on Sl(3, $\mathbb{R}$) Group

Input:
A set of Sl(3, $\mathbb{R}$) points $\{X_i\}_{i=1}^n$, $X_i \in$ Sl(3, $\mathbb{R}$)
$\epsilon > 0$
Output:
The mean point, u, on the manifold

Initialize $u = X_1$
Repeat
$$\Delta X_i = u^{-1} X_i$$
$$\Delta u = \mathrm{Rexp}\left(\frac{1}{n}\sum_{i=1}^{n} \mathrm{Rlog}(\Delta X_i)\right)$$
$$u = u\Delta u$$
Until convergence: $\|\mathrm{Rlog}\,(\Delta u)\| < \epsilon$

C. RIEMANNIAN STRESS MAJORIZATION

Although we have solved the issues of computing geodesic distances and Riemannian mean on the Sl(3, $\mathbb{R}$) manifold in section IV-A and IV-B, visualizing manifolds is still a challenging problem due to requiring algorithms that faithfully respect the non-Euclidean geometry.

In recent years, the stress majorization method deriving from multi-dimensional scaling [39]–[41] is widely adopted on graph layout [42], [43]. Based on the fact that stress majorization has an improved mathematical basis and better convergence properties [44], [45], we extend this method into the Sl(3, $\mathbb{R}$) group. Our main purpose is to provide visual representations with the geometric property of the geometric transformation group in the high dimensional space. Given a set of data points $\{X_k, k = 1, 2, \cdots, n\}$, the aim of our framework is to obtain a set of Euclidean coordinate vectors $Z = \{Z_k, k = 1, 2, \cdots, n\} \subset \mathbb{R}^m$, for $m = 2$ or $m = 3$, where Z_k is the vector corresponding to X_k. Taking advantages of Riemannian metrics, the stress function on the Riemannian manifolds can be defined as

$$\mathrm{stress}(Z) = \sum_i \sum_{j<i} w_{i,j}\left(\|Z_i - Z_j\| - d(X_i, X_j)\right)^2 \tag{27}$$

where $d(\cdot)$ represents the geodesic distance on the manifold and $X_i, X_j \in$ Sl(3, $\mathbb{R}$).

Furthermore, similarly to [45], we adopt the weight value as $w_{i,j} = d_{X_i,X_j}^{-2}$, which always generates desirable results in most cases. Moreover, if we simplify the stress function, it will be obvious that the quadratic majorization is more efficient and convenient than optimizing the original stress function. Thus, we minimize Eq.27 by calculating its supremum as follows. After expanding Eq.27, we can obtain that

$$\begin{aligned}\mathrm{stress}(Z) &= \sum_i \sum_{j<i} w_{i,j} d_{X_i,X_j}^2 + \sum_t \sum_{j<i} w_{i,j}\|Z_i - Z_j\|^2 \\ &\quad - 2\sum_i \sum_{j<i} \delta_{i,j}\|Z_i - Z_j\| \\ \text{with } \delta_{i,j} &= w_{i,j} d_{X_i,X_j} \text{ for } i, j = 1, 2, \cdots, N\end{aligned} \tag{28}$$

The first term of Eq.28 is a constant value of no importance. The second term of Eq.28 is a quadratic sum so that it may be converted to the matrix form of the weighted Laplacian L^w as follows

$$\sum_i \sum_{j<i} w_{i,j}\|Z_i - Z_j\|^2 = tr\left(Z^T L^w Z\right) \tag{29}$$

where $L_{i,j}^w = \begin{cases} -w_{i,j}, i \neq j \\ \sum_{k\neq i} w_{i,k}, i = j \end{cases}$.

The third term in Eq.28 is more intricate and cannot be easily rewritten in the matrix form. In this circumstance, we can construct the boundary function of this term as follows. Concretely, taking advantage of the Cauchy-Schwartz inequality, there exists

$$\|x\|\,\|y\| \geq x^T y, \quad \text{with equality when } x = y \tag{30}$$

Thus, given any $n \times d$ matrix U, we can obtain that

$$\begin{aligned} \|Z_i - Z_j\| \|U_i - U_j\| &\geq (Z_i - Z_j)^T (U_i - U_j), \\ \text{with equality when } U &= Z \end{aligned} \tag{31}$$

As for now, the third term can be bounded as

$$\begin{aligned} &\sum_i \sum_{j<i} \delta_{i,j} \|Z_i - Z_j\| \\ &\geq \sum_i \sum_{j<i} \delta_{i,j} (Z_i - Z_j)^T (U_i - U_j) \frac{1}{\|U_i - U_j\|} \\ &= tr\left(Z^T L^U U\right) \end{aligned} \tag{32}$$

where $L^U_{i,j} = \begin{cases} -\delta_{i,j} \left(\|U_i - U_j\|\right)^{-1}, i \neq j \\ -\sum_{k \neq i} L^U_{i,k}, i = j. \end{cases}$

Combining all these above, the supremum of the objective function can be written as

$$F^U(Z) = \sum_i \sum_{j<i} w_{i,j} d^2_{X_i,X_j} + tr\left(Z^T L^w Z\right) - 2tr\left(Z^T L^U U\right) \tag{33}$$

Thus, $F^U(Z)$ is the supremum of Eq.27 defined as

$$\text{stress}(Z) \leq F^U(Z), \quad \text{with equality when } U = Z. \tag{34}$$

Obviously, $F^U(Z)$ is a quadratic form which bounds the stress. Subsequently, we differentiate it with respect to Z and the solution of $F^U(Z)$ can be optimized as

$$L^W Z = L^U U \tag{35}$$

Or, equivalently, the optimization problem Eq.35 can be transformed to the solution of p quadratic problems based on [46], one for each axis:

$$Z_{(a)}{}^T L^W Z_{(a)} - 2 Z_{(a)}{}^T L^U U_{(a)} \tag{36}$$

The optimization of Eq.36 is convex because of the positive semi-definite property of L^W. Equivalently, the convex quadratic optimization problem can be written as,

$$\min_{Z_{(a)} \in \mathbb{R}^n} Z_{(a)}{}^T L^W Z_{(a)} + Z_{(a)}{}^T b \tag{37}$$

where $L^W \succcurlyeq 0$ and $b = -2L^U U_{(a)} \in \mathbb{R}^n$.

Finally, Eq.37 can be iteratively optimized by the conjugate gradient descent method [46] as shown in Algorithm.2.

All in all, the whole algorithm of visualizing the Sl(3, $\mathbb{R}$) group is described in Algorithm.3:

V. EXPERIMENTS

In Section III-B, we have introduced that the Lie group representations of the geometric transformation in images are 3×3 real matrices with 8 independent parameters. Because we aim to visualize the changing process of the geometric transformation of objects in images, the trajectory which reflects this process should be expressed in the three-dimensional space stemming from Sl(3, $\mathbb{R}$) $\rightarrow \mathbb{R}^3$. In practice, we obtain the geometric transformation matrices based on the implementation in [24].

Algorithm 2 Riemannian Stress Majorization on Sl(3, $\mathbb{R}$)

Input:
the initial guess $Z_{(a)} \in \mathbb{R}^n$, $1 \leq a \leq p$
$\epsilon > 0$
Output:
the coordinate matrix Z

Initialize $Z_{(a)} \in \mathbb{R}^n$
Repeat
Set gradient $\gamma = 2L^W Z_{(a)} + b$
Set step size $s = \frac{1}{2} \frac{\gamma^T \gamma}{\gamma^T L^W \gamma}$
update $Z_{(a)} = Z_{(a)} - s\gamma$
Until convergence: $\|s\gamma\| \leq \epsilon$

Algorithm 3 Visualization of the Geometric Transformation Group on Sl(3, $\mathbb{R}$)

Input:
the image set $I = \{I_1, I_2, \cdots\cdots, I_n\}$
Output:
the coordinate matrix Z

Steps
1. Compute the geometric transformation matrices from the image set I according to [24];
2. Compute the mean point, u, on the Sl(3, $\mathbb{R}$) manifold according to Algorithm.1;
3. Update all sample points on the manifold by left-multiplying the inverse of u according to Eq.26.
4. Project all updated sample points on the manifold to the unique tangent space of the identity element by virtue of Eq.18 to obtain corresponding tangent vectors;
5. Compute the coordinates in three-dimensional state transition space based on Algorithm.2;
6. Visualization process.

FIGURE 3. The original image of a stadium for simulation.

A. SYNTHETIC EXPERIMENTS

As Fig.3 shown, the original image is an aerial photography of a stadium online. In our simulation model, we assume that the aerial object is to fly at about Mach 3.5 and we obtain 50 frames of the object per second. Thus, the distance

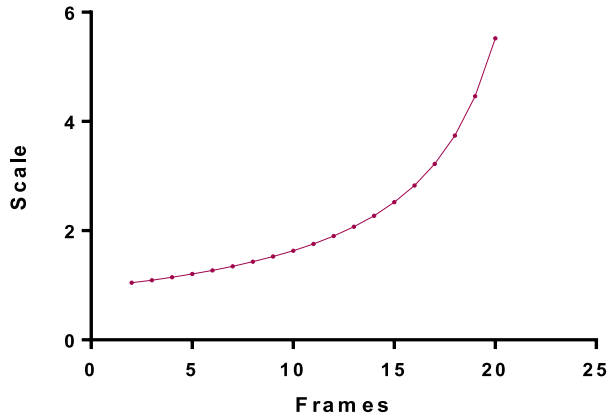


FIGURE 4. The relationship of magnified scale and frames.

that the object moves in every frame, v, can be computed as:

$$v = \frac{3.5 \times 340\text{m/s}}{50\text{frame/s}} = 23.8\text{m/frame} \tag{38}$$

In this case, the distance from the aerial object to the ground target at the j-th frame, L, can be computed as:

$$L = L_0 - v \times \text{Frame} = 560 - 23.8 \times (j-1) \tag{39}$$

where L_0 is the distance of the aerial detector to the ground target in the first frame, which means the aerial object reaches the target from the distance of 560 meters at the beginning. And Frame denotes the time period of the aerial object at the j-th frame.

In order to generate synthetic image sequences including geometric transformations, we simulate the geometric transformation during the tracking process of the aerial object and impose the scale of magnification on the original image in Fig.3 by making use of the relationship as follows:

$$s = \frac{L_0}{L} = \frac{L_0}{L_0 - v \times \text{Frame}} \tag{40}$$

where s is the scale magnification coefficient of the relationship between the j-th frame and the first frame.

As we can see, Eq.40 denotes the relationship of the scale magnification parameter between two neighbor frames, which is determined by the aerial distance of the target and the detector. This relationship is illustrated in Fig.4 and the corresponding generated images are shown in Fig.5(a). The order of image sequences is from left to right, from top to bottom and all the following order is similar. By using our visualization framework, the smooth trajectory which visualizes the process of the geometric transformation in image sequences of Fig.5(a) is exhibited in Fig.5(b). Moreover, in Fig.5(b), each number associated with the sample point corresponds to the order of image sequences.

Combing with Fig.4 and Fig.5(b), we can conclude that the curve visualizes the changing process of the geometric transformation accurately. Because the geometric transformation is produced from the relationship of Eq.40 drawn in Fig.4, the visualization curve coincides well with the tendency in Fig.4. More specifically, it is obvious that the relationship in Fig.4 increases rapidly from the 15-th frame to the 20-th frame. Correspondingly, from the point #15[2] on, the visualized trajectory also exist gradually dramatic transformations in Fig.5(b). Moreover, the distance histograms of pairwise points in the reproduced visualization space in Fig.5(d) are almost coincident with the pattern of distances between pairwise points on the original manifold in Fig.5(c), which attributes to utilizing the geometric structure of the manifold desirably.

B. EXPERIMENTS ON REAL IMAGES

1) VISUALIZATION OF AERIAL IMAGES

Recently, the aerial images obtained from drones and aerial vehicles have attracted more and more attention with the development of artificial intelligence. In this context, visualization and comprehension of geometric transformations of aerial images make significant sense. In this subsection, we use the real aerial images obtained from the visual tracking task to conduct visualization experiments.

From the whole perspective, the visualization results are shown in Fig.6. Firstly, Fig.6(a) shows the sequences of the aerial images. Fig.6(b) visualizes the trajectory of the geometric transformation computed by aerial images in Fig.6(a). The initial point with label '1' represents the first frame, and by this analogy, the final point marked as '11' denotes the last sequence based on the map $SL(3, R) \rightarrow \mathbb{R}^3$. The trajectory reflects the changing process of the geometric transformation correspondingly. For example, the distance between point #7 and point #8 is farther than its neighbors because the object goes through an obvious geometric transformation, including tremendous rotation transformation and violent distance changes. The latter five images go through an obvious transformation primarily on the rotation transformation, which leads to the obvious change of trajectory from point #8 to point #11. As the motion trajectory shown in Fig.6(b), the visualization is consistent with the changing pattern of the geometric transformation very well. Finally, comparing to the results of Fig.6(c) and Fig.6(d), we can conclude that our visualization result of Fig.6(b) has high equality because the distances of points in the visualized space preserve the original distances of pairwise points well.

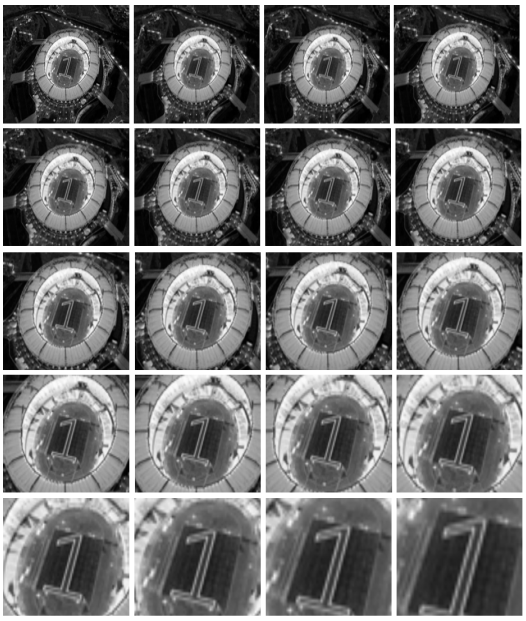
2) VISUALIZATION OF IMAGES ON VISUAL TRACKING

In section V-B1, we conduct experiments on the aerial images to visualize the geometric transformations during visual tracking. Nevertheless, not only restricted to aerial images, but also the geometric transformations of moving planar objects play an important role in the visual tracking tasks of computer vision.

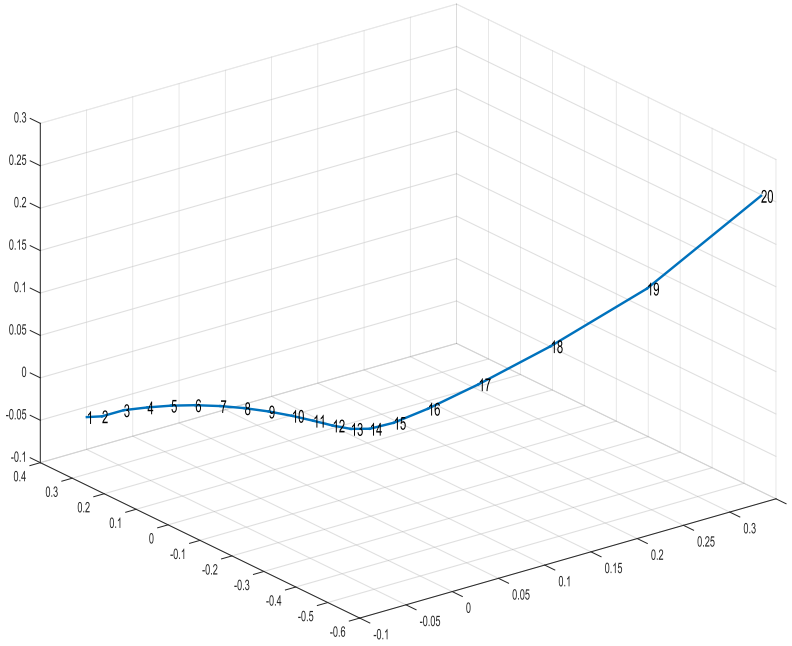
In this part, we test our method on sequences[3] of moving planar objects from [24] in order to observe whether the visualized curve reflects the projective transformation

[2]Here, # stands for the order of numbers. For example, point #15 denotes the point with the label of number 15.

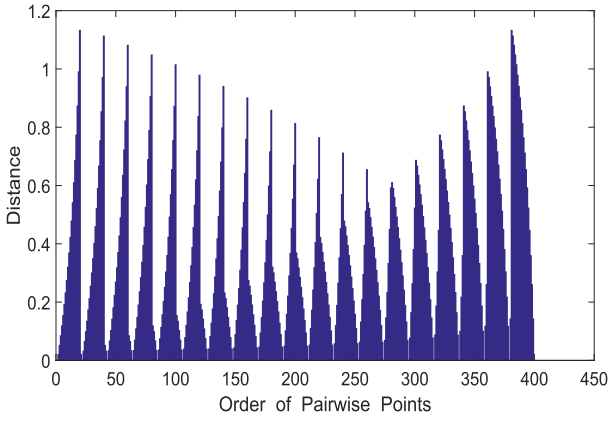
[3]The dataset can be downloaded at the website ftp://ftp-sop.inria.fr/evolution/malis/software/ESM/seq1.tar.gz



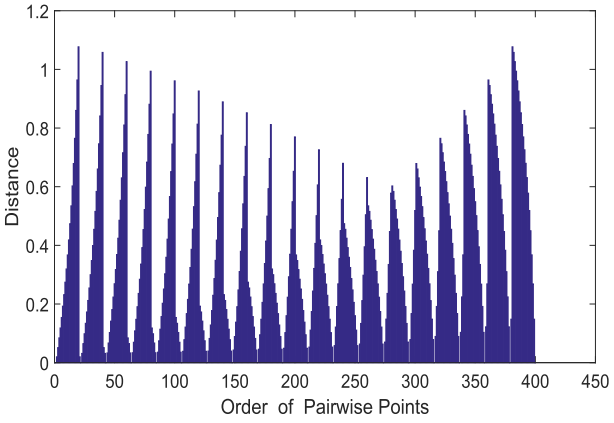
(a) Image sequences generated from Fig.3: the order is from left to right, from top to bottom.



(b) Visualization trajectory of simulated image sequences. The distances of coordinates express the extents of corresponding image geometric transformations.



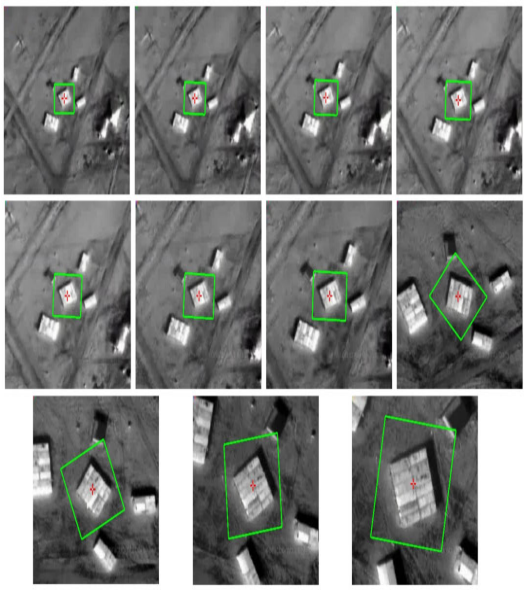
(c) Distance histogram in the original space.



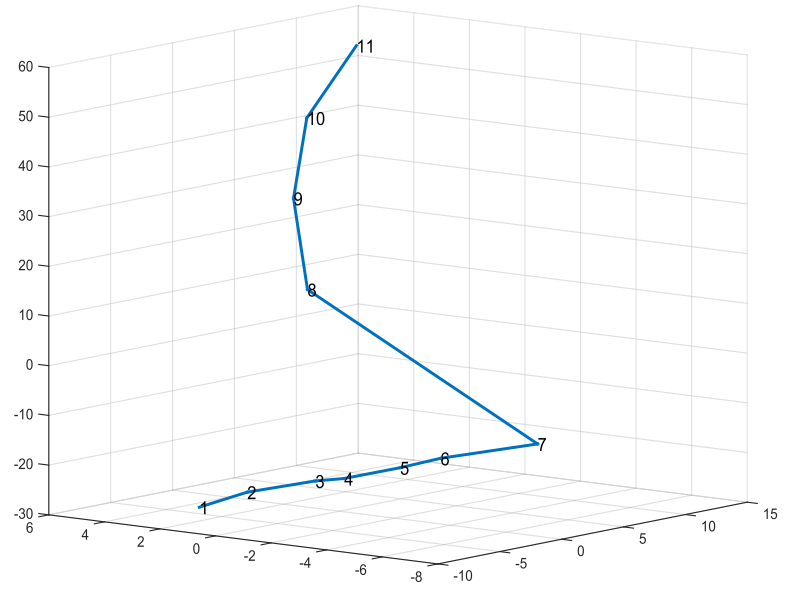
(d) Distance histogram in the visualization space.

FIGURE 5. Visualization results of the geometric transformations in the simulated image sequences: Fig.5(a) shows the experimental samples of the simulated image sequences. Fig.5(b) illustrates the visualization trajectory of the geometric transformations in Fig.5(a). Fig.5(c) draws the pattern of original distances among actual points on the manifold in Fig.5(a) and Fig.5(d) reflects the pattern of pairwise distances of the computed coordinate-points in Fig.5(b), respectively.

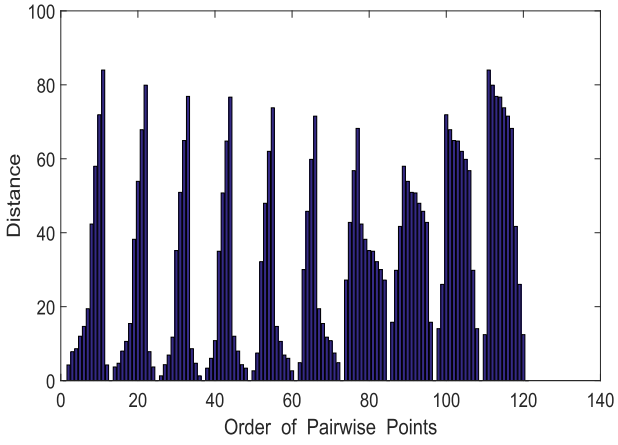
correctly if the transformation processes are more complicated relatively. For visual tidiness, we sample 20 images with 10 stepsize from the total 200 image sequences for the purpose of visualization. In this approach, the relevant sampling image sequences are displayed in Fig.7(a). From the visualization trajectory shown in Fig.7(b), we can observe that the distance between point #1 and point #2 is longer than that of point #2 and point #3. The distance between point #3 and point #4 is also longer than that of point #2 and point #3 obviously. From the perspective of transformation, analyzing the reasons for the phenomenon mentioned above are that the box between sequence #1 and sequence #2 undergoes an apparent translation in contrast to the transformation existing between frame #2 and frame #3. Analogously, the geometric transformation between sequence #3 and sequence #4 experiences an obvious rotation of large angle, which results in the increment of distance in Fig.7(b). Furthermore, the transformation between frame #11 and frame #12 goes through a wide-angle rotation as opposed to its neighbors, which leads to a long distance of points. Similarly, from point #15 to point #20 of the trajectory, we can deduce there must exist drastic geometric transformations and so it is judging from the associated image sequences in Fig.7(a). According to the comparison between Fig.7(c) and Fig.7(d), we can also observe that the distribution of points on the manifold are reconstructed perfectly in the visualization space.



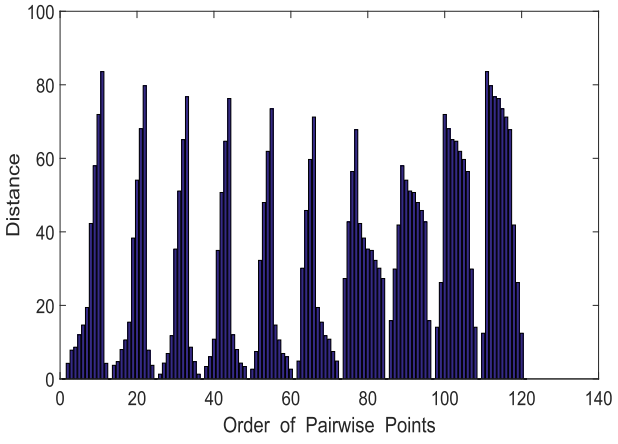
(a) The image sequences of the aerial target.



(b) Visualization trajectory of the aerial image sequences. The distances of coordinates express the extents of corresponding image geometric transformations.



(c) Distance histogram in the original space.



(d) Distance histogram in the visualization space.

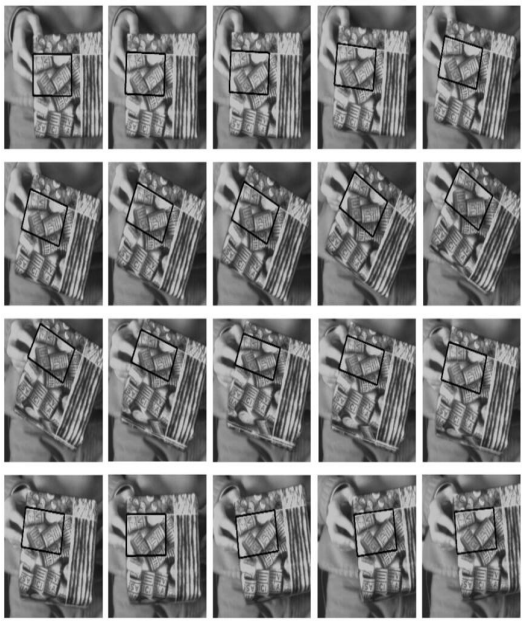
FIGURE 6. **Visualization results of the geometric transformations in the aerial image sequences: Fig.6(a) shows the experimental samples of the aerial image sequences. Fig.6(b) illustrates the visualization trajectory of the geometric transformations in Fig.6(a). Fig.6(c) reflects the pattern of original distances among actual points on the manifold in Fig.6(a) and Fig.6(d) draws the pattern of pairwise distances of the computed coordinate-points in the visualization space in Fig.6(b), respectively.**

C. EVALUATION PERFORMANCE AND DISCUSSION

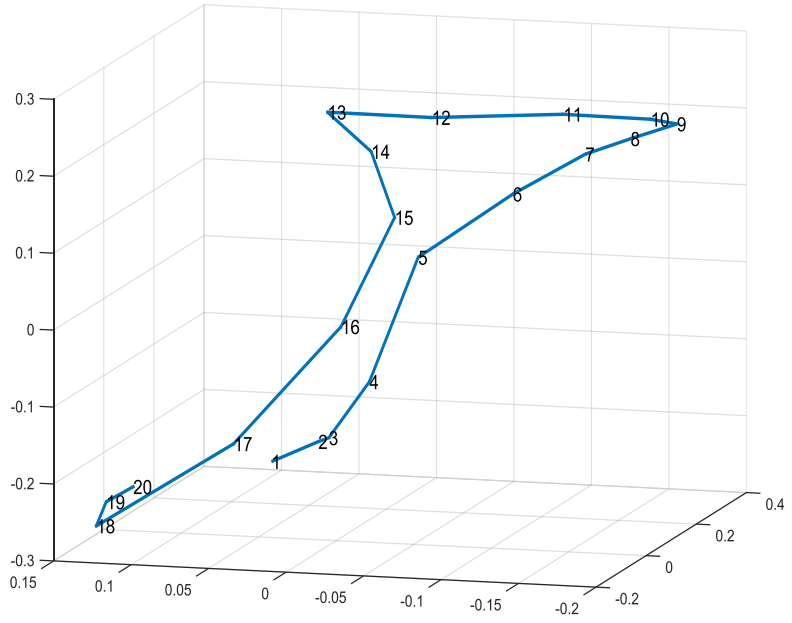
1) METHODS OF COMPARISON

In this section, to reflect the effective performance of the proposed algorithm, we first select some other state-of-the-art visualization methods for comparison:

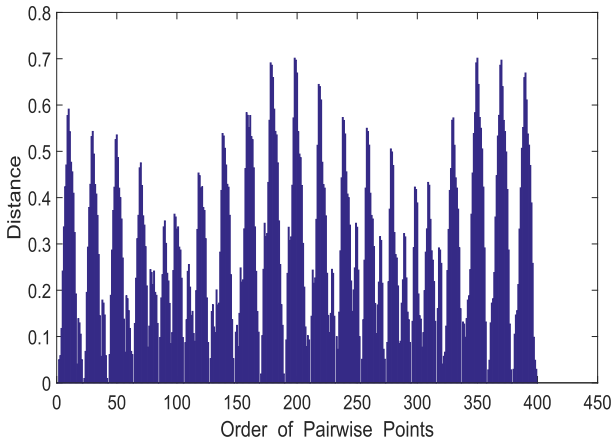
- Principal component analysis (PCA) [47]: It is the most widely used algorithm for the dimension reduction purpose, which transforms a plurality of variables into a few comprehensive variables (i.e., the principal components) through the study of the correlation matrix of the original variables or the internal structure of the covariance matrix.
- Locally linear embedding (LLE) [48]: It is one of the most classical methods of manifold learning, which makes the data of dimensionality reduction maintain the original manifold structure. This method regards each data point and its neighbors being located in a linear or approximately linear region of the manifold, which converts the global nonlinearity to the local linearity.
- Isometric Feature Mapping (Isomap) [49]: It constructs the shortest path in the nearest neighbor graph to obtain the approximate geodesic distance to replace the Euclidean distance that cannot represent the inner manifold structure, and then finds the low-dimensional coordinates of points embedded in the high-dimensional space.
- T-Distribution Stochastic Neighbor Embedding (T-SNE) [50]: It transforms the similarity between data points to the probability for visualization. In this



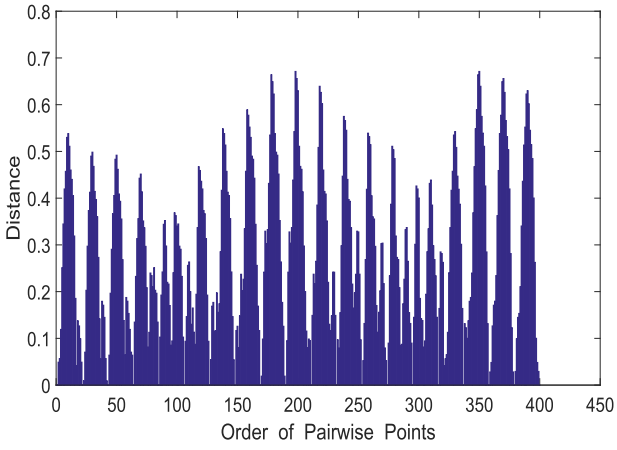
(a) The image sequences of moving object sampling with 10 stepsize.



(b) Visualization trajectory of the moving object image sequences . The distances of coordinates express the extents of corresponding image geometric transformations.



(c) Distance histogram in the original space.



(d) Distance histogram in the visualization space.

FIGURE 7. Visualization results of the geometric transformations in the image sequences of moving object: Fig.7(a) shows the experimental samples of the moving object. Fig.7(b) illustrates the visualization trajectory of the geometric transformations in Fig.7(a). Fig.7(c) is the pattern of original distances among actual points on the manifold in Fig.7(a) and Fig.7(d) is the pattern of pairwise distances of the computed coordinate-points in the visualization space in Fig.7(b), respectively.

approach, the similarity in the original space is represented by the joint Gaussian probability distribution, and the similarity of the embedded space is represented by the t-distribution. Through the optimization based on the Kullback–Leibler divergence, the visualization of similarity data in the two or three dimensional space is obtained from the perspective of probability.

2) OBJECTIVE EVALUATION INDICATORS OF VISUALIZATION

Next, we provide a systematic evaluation of these visualization methods on different quantitative and qualitative measurements in respect of accuracy, utility, and efficiency. Note that it is difficult to find some generalized and widely-used measurements of visualization approaches taking account of the subjective particularity and the deficient literature in this research field. Facing this issue, we fairly define some objective evaluation indicators to estimate the quality of the visualization results combing with relevant literature [51]–[53].

More specifically, we propose three objective evaluation indicators including:

-1) accuracy (ACC);
-2) number of effective points (NEP);
-3) variance mutual information (VMI).

- The accuracy of visualization is defined to measure the proportion that the reconstruction preserves from the original pairwise distances. This measurement is the key

indicator of our measurements, which can be computed as:

$$\mathrm{ACC} = 1 - \frac{\sum_{i=1}^{n}\sum_{i\neq j}\left(\left\|Z_i - Z_j\right\| - d\left(X_i - X_j\right)\right)^2}{\sum_{i=1}^{n}\sum_{i\neq j} d^2\left(X_i - X_j\right)} \tag{41}$$

where $d\left(X_i - X_j\right)$ represents the distance of point X_i and X_j, $\left\|Z_i - Z_j\right\|$ denotes the corresponding distance on the visualization space and n is the number of sample points.

- The NEP reflects the number of effective points in the total pairwise points, which is defined as

$$\mathrm{NEP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{42}$$

where TP means the number of effective points and FP corresponds to the number of invalid points. In order to measure the validation of pairwise points, we also define a parameter, α, in the following way:

$$\mathrm{NEP}\left(i, j\right) = \begin{cases} 1, & \text{if } \left|\dfrac{\left\|Z_i - Z_j\right\| - d\left(X_i - X_j\right)}{d\left(X_i - X_j\right)}\right| \leq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

where $\mathrm{NEP}(i, j)$ with $1 \leq i, j \leq n$ represents whether the pairwise samples of point #i and point #j belong to the effective points or not. Furthermore, the value of parameter α is selected within the range of $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ in our experiments.

- The VMI computes the variance of the point distances in the visualization results, which is given by

$$\mathrm{VMI} = \mathrm{var}(\mathrm{error}) \tag{44}$$

where $\mathrm{error} = (e_1, e_2, \cdots, e_N)$ represents the error vector and $e_m = \left|\left\|Z_i - Z_j\right\| - d\left(X_i - X_j\right)\right|$ with $1 \leq m \leq N$ corresponds to the residual error of point #i and point #j. N is the number of pairwise points and var$(\cdot)$denotes the function to compute the variance.

3) EVALUATION PERFORMANCE

In Section.V-A and Section.V-B, we exhibit the visualization trajectories of image geometric transformations on different datasets. In this part, we further provide numerical evaluations of objective indicators defined in SectionV-C2 to measure quantitative performances of our method and other visualization methods for comparison.

The objective evaluation indicators of visualization on the stadium sequences are shown in Table.1. The corresponding blue numbers represent the best experimental results of all methods. From the ACC results in Table.1, we can observe that although other comparison algorithms have achieved desirable accuracies, our method still outperforms them than at least 7%. Especially, the ACC of our method is more than 17% higher compared to T-SNE. For the NMI, our method also generates the best performance. As for the NEP with various α, all the methods can obtain relatively good results when α is larger than 0.3. Actually, according to the definition of NEP, the higher amount NEP achieves with the lower value of α, the better performance the approach will possess for visualization. In this criterion, our method always gets desirable performances when α is in the range of small values. When $\alpha = 0.05$, our algorithm obtains at least 21% increasement than other visualization methods. When $\alpha = 0.1$, our method also outperforms the best results of comparison approaches about 5%.

Next, the objective evaluation indicators of visualization on the aerial image sequences are shown in Table.2. From the whole perspective, our approach performs best in terms of all relevant indicators. The ACC of our method is at least 4% higher than other methods, and the values of NEP with small α are also nearly 25% higher than the other four comparison methods. Particularly, our NMI is reduced in one order of magnitude compared to other algorithms. Analogously, we can obviously observe that our method also exhibits desirable evaluation performances in the experiments of SectionV-B1 from the experimental results reported in Table.3. The details of the corresponding analyses based on Table.3 will be given in the next part.

4) PERFORMANCE WITH DIFFERENT NUMBERS OF IMAGE SEQUENCES

In this section, we aim to test the performance of our method when the number of experimental image sequences used is increased.

In practice, we conduct extensive experiments on the same dataset in SectionV-B1 to provide the systematic behavior of the proposed approach under different number of samples. In our experiments, we increase the number of experimental image sequences, n, from 20 to 200. More specifically, the value of n is selected in the range of $\{20, 50, 80, 120, 150, 200\}$. The corresponding objective evaluation indicators of visualization under enhanced number of image sequences are reported in Table.3. From these results, we can conclude that our method has strong robustness even if the number of experimental image sequences is raised in the one order of magnitude. From the measurement of accuracy, the proposed algorithm always keeps in the high-level performances that are upper than 96%. As for the NMI, although this indicator grows obviously with n, it is still lower than 0.1 when n equals to 200, which means our method is stable with various numbers of samples. Similarly, the indicators of NEPs with different α also perform acceptably under the large amount of n.

All in all, the visualization results in Fig.5-Fig.7 and the quantitative evaluation measurements in Table.1-Table.3 illustrate that our method not only leads to the high consistency with the image geometric transformation, but also obtains ranking behaviors on objective estimation indicators, which demonstrates the effectiveness and robustness of the proposed algorithm.

TABLE 1. Objective evaluation indicators of visualization on the stadium sequences in SectionV-A.

Indicator / Method	ACC	NEP						NMI
		($\alpha = 0.05$)	($\alpha = 0.1$)	($\alpha = 0.2$)	($\alpha = 0.3$)	($\alpha = 0.4$)	($\alpha = 0.5$)	
PCA	0.8997	0.75	0.96	1.00	1.00	1.00	1.00	0.0344
LLE	0.8923	0.31	0.49	0.775	0.90	0.955	0.975	0.1644
Isomap	0.8944	0.75	0.95	1.00	1.00	1.00	1.00	0.0326
T-SNE	0.8004	0.545	0.76	0.865	0.92	1.00	1.00	0.0769
Our Method	0.9706	0.965	1.00	1.00	1.00	1.00	1.00	0.0129

TABLE 2. Objective evaluation indicators of visualization on the aerial image sequences in SectionV-B1.

Indicator / Method	ACC	NEP						NMI
		($\alpha = 0.05$)	($\alpha = 0.1$)	($\alpha = 0.2$)	($\alpha = 0.3$)	($\alpha = 0.4$)	($\alpha = 0.5$)	
PCA	0.9570	0.5455	0.7273	0.8364	0.8909	0.9273	0.9455	1.8279
LLE	0.9376	0.4545	0.9818	0.9818	1.00	1.00	1.00	1.884
Isomap	0.9286	0.5091	0.7091	0.8545	0.9273	1.00	1.00	1.9646
T-SNE	0.9317	0.6909	0.76	0.865	0.92	1.00	1.00	1.9697
Our Method	0.9965	1.00	1.00	1.00	1.00	1.00	1.00	0.1972

TABLE 3. Performance of our method under different number of image sequences in SectionV-B2.

Indicator / Number of Image Sequence n	$n = 20$	$n = 50$	$n = 80$	$n = 120$	$n = 150$	$n = 200$
ACC	0.9956	0.9932	0.9861	0.9769	0.9738	0.9673
NMI	0.0105	0.0287	0.0206	0.0328	0.0419	0.0588
NEP ($\alpha = 0.05$)	1.00	0.9784	0.8603	0.8264	0.8048	0.7699
NEP ($\alpha = 0.1$)	1.00	1.00	0.975	0.960	0.9349	0.8850
NEP ($\alpha = 0.2$)	1.00	1.00	1.00	0.9764	0.9947	0.9212
NEP($\alpha = 0.3$)	1.00	1.00	1.00	1.00	1.00	0.9805
NEP ($\alpha = 0.4$)	1.00	1.00	1.00	1.00	1.00	1.00
NEP ($\alpha = 0.5$)	1.00	1.00	1.00	1.00	1.00	1.00

VI. FURTHER DISCUSSION

A. COMPUTATIONAL COMPLEXITY

As mentioned in Section IV, the main computational cost of our proposed algorithm depends on the procedures to solve the convex quadratic optimization problem by the conjugate gradient descent method. In each iteration, the optimization involves matrix-vector multiplication between matrices of size $n \times n$ and $n \times 1$, which takes n^2 flops. Because the optimization problem Eq.35 is transformed to the solution of p quadratic problems, the optimization sums to pn^2 flops for each iteration. Let k denote the number of iterations.

Overall, our method mainly demands kpn^2 extra flops, which is linear in p ($p \leq 3$) and causes all the steps to have affordable computational complexity.

B. CONVERGENCE

For the related mathematical analysis and discussion of the optimization algorithm of Section IV, readers can refer to the literature [44], [45]. Furthermore, we conduct the additional experiment to perform the convergence behavior of our method. Figure 8 illustrates the typical convergence behavior of the proposed approach in the experiments of Section V-A.

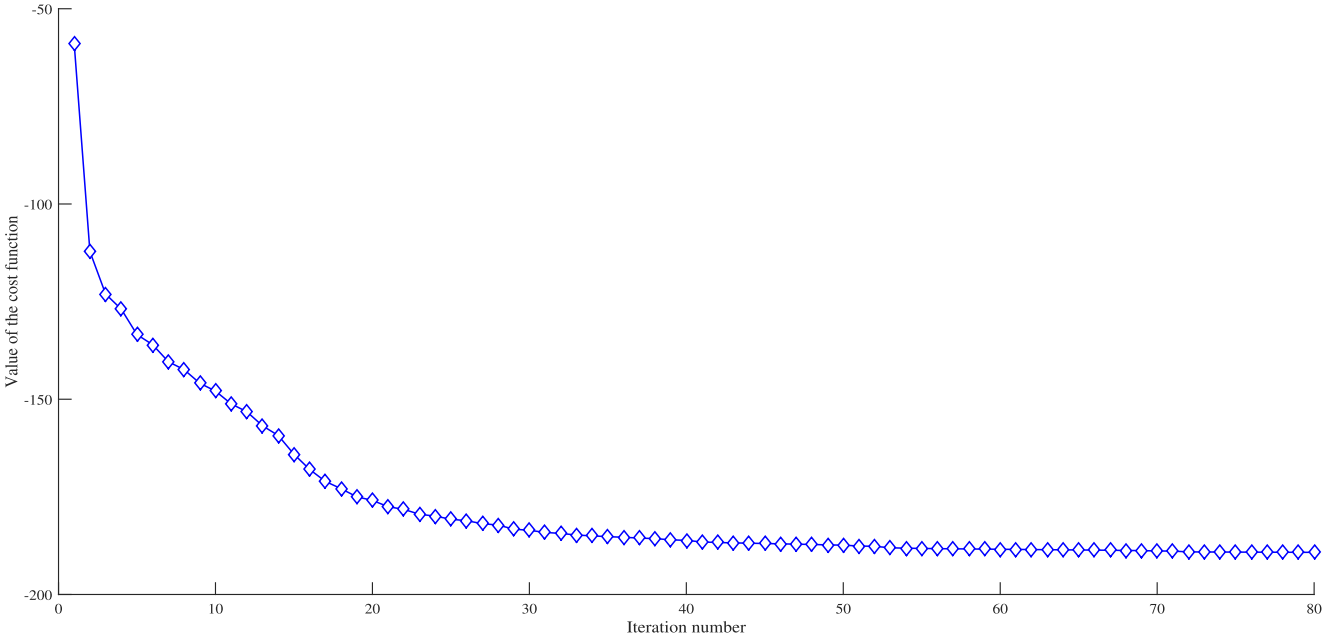


FIGURE 8. The convergence behavior of the proposed method.

In practice, the algorithm generally converges rapidly in fewer than 80 iterations.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a new framework to visualize the changing process of geometric transformations in images based on Riemannian manifold. In particular, we define a distance metric to compute the geodesic length on the $\mathrm{Sl}(3,\mathbb{R})$ group on account of the fact that the lack of analytic expressions of Riemannian logarithmic map on the $\mathrm{Sl}(3,\mathbb{R})$ manifold is the notorious obstacle for visualization. Once the hard-core issue of the distance metric is solved, the geometric transformations in images are further visualized based on the Riemannian stress majorization. The experimental results on diverse visual data demonstrate the effectiveness and performance of our approach. Not only is our framework valuable for analyzing motion groups of visual tasks, but also it will provide experience for other applications including medical imaging science and brain-computer interface.

In the future, we plan to extend our research in two directions. Firstly, we will explore the possibility of creating deep representations of the manifold visualization with deep learning technology for better performance. Secondly, we will focus on visualizing other types of manifolds that commonly occur in visual tasks, such as rigid motion estimation on a special Euclidean group SE(3) [54], [55], rotation group SO(3) [56], [57] or subspace descriptors on the Grassmann manifold [22], [53], [58], [59].

REFERENCES

[1] Y. M. Lui, "Advances in matrix manifolds for computer vision," *Image Vis. Comput.*, vol. 30, nos. 6–7, pp. 380–388, Jun. 2012.
[2] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee, "A geometric particle filter for template-based visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 625–643, Apr. 2014.
[3] C. Li, Z. Shi, Y. Liu, T. Liu, and L. Xu, "Efficient and robust direct image registration based on joint geometric and photometric lie algebra," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6010–6024, Dec. 2018.
[4] R. Li and R. Chellappa, "Group motion segmentation using a spatio-temporal driving force model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2038–2045.
[5] D. Lin, E. Grimson, and J. Fisher, "Learning visual flows: A lie algebraic approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 747–754.
[6] T. Liu, Z. Shi, Y. Liu, and C. Li, "Visualization of the geometric transformation group based on the riemannian metric," in *Proc. 3rd Int. Conf. Syst. Inform. (ICSAI)*, Nov. 2016, pp. 930–935.
[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[9] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," Dec. 201, *arXiv:1612.08242* [Online]. Available: https://arxiv.org/abs/1612.08242
[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[13] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," Dec. 2013, *arXiv:1312.6034*. [Online]. Available: https://arxiv.org/abs/1312.6034
[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 818–833.
[15] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. J. Ackel, U. Muller, P. Yeres, and K. Zieba, "Visual back prop: Efficient visualization of CNNs for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.
[16] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2414–2423.

[17] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4990–4998.
[18] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 17–32.
[19] M. Harandi, M. Salzmann, and R. Hartley, "Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 48–62, Jan. 2017.
[20] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 720–729.
[21] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on Grassmann manifold with application to video based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 140–149.
[22] T. Liu, Z. Shi, and Y. Liu, "Joint normalization and dimensionality reduction on grassmannian: A generalized perspective," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 858–862, Jun. 2018.
[23] E. Begelfor and M. Werman, "How to put probabilities on homographies," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1666–1670, Oct. 2005.
[24] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing," *Int. J. Robot. Res.*, vol. 26, no. 7, pp. 661–676, Jun. 2007.
[25] S. Lee, M. Choi, H. Kim, and F. C. Park, "Geometric direct search algorithms for image registration," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2215–2224, Sep. 2007.
[26] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, 2007.
[27] J. M. Lee, "Smooth manifolds," in *Introduction to Smooth Manifolds*. New York, NY, USA: Springer, 2013. pp. 1–31.
[28] L. P. Eisenhart, *Riemannian Geometry*. Princeton, NJ, USA: Princeton Univ. Press, 2016.
[29] W. Rossmann, *Lie Groups: An Introduction Through Linear Groups*, vol. 5. London, U.K.: Oxford Univ. Press, 2002.
[30] A. Baker, *Matrix Groups: An Introduction to Lie Group Theory*. Springer, 2012.
[31] I. M. Gelfand and R. A. Silverman, *Calculus of Variations*. Chelmsford, MA, USA: Courier Corporation, 2000.
[32] T. Fletcher and R. Whitaker, "Riemannian metrics on the space of solid shapes," in *Proc. 1st MICCAI Workshop Math. Found. Comput. Anatomy, Geometrical, Stat. Registration Methods Model. Biol. Shape Variability*, 2006, pp. 47–57.
[33] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.
[34] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Commun. pure Appl. Math.*, vol. 30, no. 5, pp. 509–541, Sep. 1977.
[35] E. Begelfor and M. Werman, "Affine invariance revisited," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, vol. 2, pp. 2087–2094.
[36] R. Ferreira and J. Xavier, "Hessian of the Riemannian squared distance function on connected locally symmetric spaces with applications," in *Proc. 7th Portuguese Conf. Autom. Control*, Sep. 2006, vol. 2, pp. 1–6.
[37] R. Subbarao and P. Meer, "Nonlinear mean shift over Riemannian manifolds," *Int. J. Comput. Vis.*, vol. 84, no. 1, p. 1, Aug. 2009.
[38] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Physica-Verlag HD, 2010, pp. 177–186.
[39] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. Boston, MA, USA: Chapman, 2000.
[40] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.
[41] I. Borg and P. Groenen, "Modern multidimensional scaling: Theory and applications," *J. Educ. Meas.*, vol. 40, no. 3, pp. 277–280, Sep. 2003.
[42] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Proc. Int. Symp. Graph Drawing*. Berlin, Germany: Springer, 2004.
[43] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, Apr. 1989.
[44] J. D. Leeuw, "Convergence of the Majorization method for multidimensional scaling," *J. Classification*, vol. 5, no. 2, pp. 163–180, Sep. 1988.
[45] T. Dwyer, Y. Koren, and K. Marriott, "Stress majorization with orthogonal ordering constraints," in *Proc. Int. Symp. Graph Drawing*. Berlin, Germany: Springer, 2005.
[46] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore MD, USA: JHU Press, 2012.
[47] G. H. Dunteman, *Principal Components Analysis*, vol. 69. Oaks, CA, USA: Sage, 1989.
[48] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[49] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[50] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[51] Y. Zhu, "Measuring effective data visualization," in *Proc. Int. Symp. Visual Comput.* Berlin, Germany: Springer, 2007, pp. 652–661.
[52] Z. Daoqiang and C. Shiguo, "Experimental comparisons of semi-supervised dimensional reduction methods," *J. Softw.*, vol. 22, no. 1, pp. 28–43, 2011.
[53] T. Liu, Z. Shi, and Y. Liu, "Kernel sparse representation on grassmann manifolds for visual clustering," *Proc. SPIE*, vol. 57, no. 5, May 2018, Art. no. 053104.
[54] A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2017, pp. 173–180.
[55] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
[56] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1243–1252.
[57] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," Georgia Inst. Technol., Atlanta, GA, USA, 2015.
[58] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson, "Extrinsic methods for coding and dictionary learning on Grassmann manifolds," *Int. J. Comput. Vis.*, vol. 114, nos. 2–3, pp. 113–136, Sep. 2015.
[59] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Localized LRR on Grassmann manifold: An extrinsic view," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2524–2536, Oct. 2018.

TIANCI LIU was born in Fushun, Liaoning, China. He received the B.S. degree in mathematics and applied mathematics from the China University of Petroleum-Beijing, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences (CAS), Shenyang, China, and the University of Chinese Academy of Sciences (UCAS), Beijing. His current research interests include manifold learning, deep learning, Riemannian optimization, and pattern recognition.

ZELIN SHI received the B.S. degree in mathematics and the M.S. degree in computer application from Xi'dian University, Xi'an, China, in 1987 and 1990, respectively, and the Ph.D. degree in mechanical and electronic engineering from the Graduate School of Chinese Academy of Sciences, Beijing, China, in 2004. He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include imaging, image processing, pattern recognition, and optoelectronic systems.

YUNPENG LIU received the Ph.D. degree in pattern recognition and machine intelligence from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2010, where he is currently a Professor. His research interests include object detection, deep learning, visual tracking, and pattern recognition based on Lie group and Riemannian manifold.