Affine Interpolation in a Lie Group Framework

SUMUKH BANSAL, Dhirubhai Ambani Institute of Information and Communication Technology, India
ADITYA TATU, Dhirubhai Ambani Institute of Information and Communication Technology, India

Affine transformations are of vital importance in many tasks pertaining to motion design and animation. Interpolation of affine transformations is non-trivial. Typically, the given affine transformation is decomposed into simpler components which are easier to interpolate. This may lead to unintuitive results, while in some cases, such solutions may not work. In this work, we propose an interpolation framework which is based on a Lie group representation of the affine transformation. The Lie group representation decomposes the given transformation into simpler and meaningful components, on which computational tools like the exponential and logarithm maps are available in closed form. Interpolation exists for all affine transformations while preserving a few characteristics of the original transformation. A detailed analysis and several experiments of the proposed framework are included.

CCS Concepts: • **Computing methodologies** → *Animation.*

Additional Key Words and Phrases: Affine transformation interpolation, Lie groups, Lie Bodies.

ACM Reference Format:
Sumukh Bansal and Aditya Tatu. 2019. Affine Interpolation in a Lie Group Framework. *ACM Trans. Graph.* 38, 4, Article 71 (July 2019), 16 pages. https://doi.org/10.1145/3306346.3322997

1 INTRODUCTION

3D affine transformations play a pivotal role in many applications pertaining to computer vision, computer graphics and geometry processing. The need to interpolate affine transformations arises ubiquitously in applications involving animation design [Shoemake 1992; Whited et al. 2010], inverse kinematics [Der et al. 2006; Sumner et al. 2005], motion estimation and averaging [Zefran and Kumar 1998], image morphing, and robotics [Selig 2010]. It is well known that the set of matrices $\mathbb{T}$ representing affine transformations forms a Lie group[Ochiai and Anjyo 2013], interpolation on which is non-trivial. Moreover, as far as possible, the interpolated transformation should preserve the properties of the original affine transformation, such as orthogonality and area preservation.

Typical approaches decompose affine transformations into rotational and shear/scale components, after which each component is handled separately. Steady Affine Motion (SAM), a scheme proposed in [Rossignac and Vinacua 2011] for affine interpolation is found to maintain several desired properties. But in cases exhibiting large shear and rotation, the SAM interpolation does not exist.

The proposed approach decomposes any given 3D invertible, orientation-preserving affine transformation[1] into a series of intuitive transformations (each coming from a Lie group) needed to deform a tetrahedron into a fixed canonical tetrahedron. This gives a Lie group representation of the given affine transformation. Thus, an orientation-preserving 3D affine transformation can be represented as a mapping between two specific tetrahedrons, and conversely, a mapping between two oriented tetrahedrons with given correspondence as a unique 3D orientation-preserving affine transformation. This is a generalization of the Lie Bodies representation of 3D triangular meshes introduced by Freifeld *et al.* [Freifeld and Black 2012]. The Lie Bodies approach represents each triangle of a mesh via a specific set of transformations needed to deform a corresponding triangle in a given template mesh to the triangle under consideration. The approach proposed in this paper represents any orientation-preserving 3D affine transformation as a decomposition into three components: a 3D rigid transformation, uniform scaling, and a specific 3D-shear, refer to Figure 1. The interpolation of the affine transformation is obtained using interpolation of the three components. The advantage of this decomposition is that closed-form solutions are known for interpolations of the three components. Moreover, several properties of the original affine transformation are preserved by the proposed scheme.

To summarize, the contributions of the paper are:

(1) The proposed approach interpolates any orientation-preserving 3D affine transformation, unlike the state-of-the-art approach in [Rossignac and Vinacua 2011].
(2) We provide a detailed analysis of the proposed interpolation scheme and show that it has several desirable properties like: (a) preserves isometry, (b) preserves volume, and (c) yields a monotonic change in the volume.
(3) The proposed interpolation scheme can also interpolate two tetrahedrons (and as a special case, triangles) related by any orientation-preserving 3D affine transformation. The interpolation is unique given a correspondence and a vertex ordering. Variation with respect to vertex ordering is analyzed in detail in the paper.

In the next section, a review of the related work is provided. In Section 3, we provide details of the proposed representation of an affine transformation/tetrahedron, along with a discussion on its existence and uniqueness. Section 4 describes the proposed interpolation algorithm and properties of the given affine transformation that are preserved by the interpolations obtained. Two important invariance properties related to our approach are analyzed in Section 5, followed by experiments and results in Section 6. We conclude the paper in Section 7.

Authors' addresses: Sumukh Bansal, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujrat, India, sumukhbansal@gmail.com; Aditya Tatu, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India, aditya_tatu@daiict.ac.in.

0730-0301/2019/7-ART71 $15.00
https://doi.org/10.1145/3306346.3322997

[1] An affine transformation $\mathbb{T} : \mathbb{R}^n \to \mathbb{R}^n$ is *orientation-preserving* if $det(\mathbb{T}) > 0$.

2 RELATED WORK

There are two scenarios where the proposed approach is useful, (a) when two objects $A(0)$ and $A(1)$, also referred to as source and target, (as meshes, point clouds or triangles in 3D) are related by a unique orientation-preserving 3D affine transformation and (b) when an orientation-preserving 3D affine transformation $\mathbb{T}$ is given. In the former, intermediate objects $A(t)$, $0 < t < 1$ need to be computed, while in the latter intermediate affine transformations $\mathbb{T}(t)$, $0 < t < 1$, such that $\mathbb{T}(0) = I$ and $\mathbb{T}(1) = \mathbb{T}$ need to be computed. We first deal with the former case, and discuss ways to map the latter into the former, later in the paper.

The set of affine transformations forms a non-linear manifold and is not compact. Closed-form expressions of differential geometric tools like the exponential map are also not available. Interpolating affine transformations is typically carried out by decomposing the given transformations into simpler transformations over which interpolation is well understood. The decomposition itself may not be intuitive, and each decomposition comes with its pros and cons. A popular decomposition consists of rotation with scale and/or shear transformations. Interpolation of rotations (SO(3)) is a well-studied problem, often solved using quaternions or matrix exponential and log maps.

In [Shoemake 1992], Shoemake *et al.* proposes to decompose the affine transformation, $\mathbb{T}$, using polar decomposition $\mathbb{T} = QS$, where Q and S are rotation and stretch components (stretch being represented by symmetric positive definite matrices). Polar decomposition was found to be more stable than SVD decomposition (which is costly to compute and sensitive to small perturbations) and QR decomposition (which is stable under small perturbations but produces unintuitive rotational components). Interpolation of the rotational component is achieved using the SLERP method for quaternions [Shoemake 1985], while the stretch component is interpolated linearly. For 2D affine transformations, [Kaji et al. 2012] use polar decomposition followed by exponential maps on rotations and symmetric positive-definite matrices to interpolate the rotation and stretch components, respectively. The limitation of polar decomposition comes in form of its inability to handle shear directly. A similar approach to model non-rigid deformation of meshes is used in [Alexa et al. 2000].

In [Alexa 2002] the use of matrix exponential and log map was advocated. The matrix exponential and log map used are not in closed form and a solution based on Taylor series approximation is used. In [Sumner et al. 2005], a combination of two approaches is used. The affine transformation is decomposed into rotation and stretch transformations, and while the rotation transformation is interpolated using closed-form matrix exponential and log map based on Rodrigues's formula [Murray et al. 1994], the stretch component is interpolated linearly.

A direct interpolation of a given 3D affine transformation is introduced by defining a matrix exponential and log map in [Rossignac and Vinacua 2011]. The maps defined are given in closed-form, and are computationally efficient to implement. The interpolated transformations are also shown to preserve important properties like isometry and volume. However, in cases exhibiting large shear and rotation, the solution does not exist.

Similar cases arise in the context of skinning where the blending of transformations is involved. Linear blend skinning (LBS) [Magnenat-Thalmann et al. 1988] and other linear methods are known to produce candy-wrapper artifacts. Decomposition based linear blending approaches are also failure-prone as the linear blending of rotations does not necessarily produce a rotation. To alleviate limitations of linear methods, non-linear methods are proposed, where the rotation transformations are treated using quaternions [Kavan and Žára 2005; Shoemake 1985], while for Euclidean transformations dual quaternions were introduced [Kavan et al. 2008]. For interpolation, the SLERP algorithm [Shoemake 1985] is used which is iterative in nature, but for practical purposes, an approximation is found to be sufficient. While the quaternion based approaches work well for relatively large rotations, they are not well suited to model non-rigid deformation, for quaternions can only capture euclidean/rigid transformations. For a detailed survey refer to [Jacobson et al. 2014].

In [Freifeld and Black 2012], Freifeld *et al.* propose a Lie group based mesh representation for triangular meshes, henceforth mentioned as *Lie Bodies*, where each triangular face of the mesh is represented by a set of transformations needed to deform a corresponding triangle on a template mesh to the triangle under consideration.

The approach proposed in this paper decomposes a given 3D orientation-preserving affine transformation into a series of simpler transformations (each from a Lie group) needed to deform a tetrahedron to a given fixed tetrahedron, analogous to the Lie Bodies framework. While the difference between our representation and the one used in the Lie Bodies framework may appear minor, it is important to note that the Lie Bodies framework provides a representation of triangles and not tetrahedrons. Thus, it may be used for computing interpolations between triangles (and hence a subset of orientation-preserving 3D affine transformations), but not for interpolating all possible orientation-preserving 3D affine transformations. Also, the translation component of the transformation is ignored in the Lie Bodies framework, as the framework is proposed for computing shape variations and shape statistics of triangular meshes, which need to be translation-invariant. However, in applications like designing motions or interpolating transformations, translation plays a crucial role. Approaches that handle translation and the linear component of the affine transformation separately [Alexa 2002; Shoemake 1992; Sumner et al. 2005], fail to capture an intuitive interpolation path. We also prove that our interpolation scheme has several desirable properties like isometry preservation, volume preservation and monotonicity of volume, and analyze its invariance properties in terms of choice of canonical tetrahedron and vertex orderings. These theoretical contributions are missing (for planar transformations) in [Freifeld and Black 2012].

In the next section, we present the proposed representation of a tetrahedron with respect to a chosen canonical tetrahedron, which can also be used to represent the orientation-preserving 3D affine transformation between the two tetrahedrons.

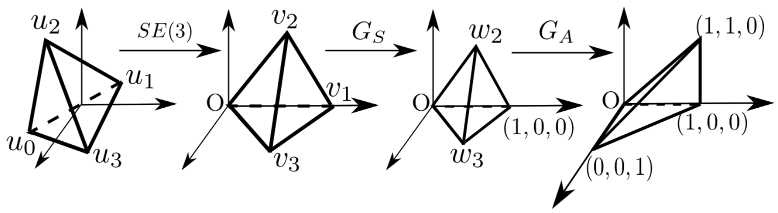

Fig. 1. Lie group representation of a tetrahedron. (left to right): (1) A tetrahedron with vertices (u_0, u_1, u_2, u_3) (in this order) is transformed using $E = [R\ \ d] \in SE(3)$ to a tetrahedron with vertices ($\mathbf{0}$, v_1, v_2, v_3) such that the face ($\mathbf{0}$, v_1, v_2) lies in the xy plane, edge ($\mathbf{0}$, v_1) aligns with the positive x axis, y coordinate of v_2 and z coordinate of v_3 are positive. (2) A 3D uniform scaling with scaling factor $s = 1/||v_1||$ is applied such that the length of the first edge vector becomes 1, to obtain the tetrahedron with vertices ($\mathbf{0}$, (1, 0, 0), w_2, w_3). (3) Finally, a 3D shear transformation A which leaves the x-axis unchanged, maps vertex w_2 to the point (1, 1, 0), and vertex w_3 to the point (0, 0, 1) is applied in order to obtain the canonical tetrahedron. The tetrahedron can thus be represented as the triplet (E, A, s) with respect to the canonical tetrahedron.

3 REPRESENTATION OF A TETRAHEDRON USING TRANSFORMATIONS

In this paper, we will denote the $n \times n$ identity matrix as I_n, a vector of n-ones or zeros by $1_n, 0_n$ respectively, and the transpose of a matrix A by A^T. We now describe the representation of a tetrahedron in $\mathbb{R}^3$. We call the tetrahedron formed by vertices $(0,0,0),(1,0,0),(1,1,0),(0,0,1)$ (in this order), a *canonical tetrahedron.* Any non-degenerate[2] tetrahedron with vertices u_0, u_1, u_2, u_3, (in this order) and same orientation[3] as the canonical tetrahedron, can be deformed into the canonical tetrahedron using a sequence of transformations described below. Refer to Figure 1 for a visualization of the process.

(1) Rigid transformation in $\mathbb{R}^3$: The rotation component R of the rigid transformation aligns: (a) face (u_0, u_1, u_2) to the xy plane, with u_2 being mapped to a point with positive y coordinate, and (b) edge (u_0, u_1) to the positive x-axis. Due to the non-degeneracy and same orientation assumption, the vertex u_3 will get mapped to a point with positive z-coordinate. The translation vector d shifts the point Ru_0 to the origin. Let us denote the new vertices by $(\mathbf{0}, v_1, v_2, v_3)$.
(2) Uniform scaling: An element of the Lie group $\mathbb{R}^+$, where the group operation is the usual multiplication between real numbers (henceforth denoted by G_S), uniformly scales the tetrahedron by $s = \frac{1}{||v_1||}$ so that the vertex v_1 is mapped to the point $(1,0,0)$. Let us denote the vertices of the resulting tetrahedron by $(\mathbf{0}, (1,0,0), w_2, w_3)$.
(3) Shear: A 3D transformation that preserves the side joining vertices $\mathbf{0}$ and $(1,0,0)$, while aligning the vertex w_2 with the point $(1,1,0)$, and vertex w_3 with the point $(0,0,1)$. Borrowing notations from [Freifeld and Black 2012], we denote this subset of transformations as $G_A := \{A \in GL(3) \mid A[1\ 0\ 0]^T = [1\ 0\ 0]^T, xy \text{ plane is an invariant subspace of } A, A_{22}, A_{33} > 0\}$,

or

$$G_A := \left\{ \begin{pmatrix} 1 & \alpha_1 & \alpha_3 \\ 0 & \alpha_2 & \alpha_4 \\ 0 & 0 & \alpha_5 \end{pmatrix} \mid \alpha_i \in \mathbb{R}, i = 1, \ldots, 5, \alpha_2, \alpha_5 \geq 0 \right\}.$$

Thus, each tetrahedron can be deformed into the canonical tetrahedron using the transformations described above. It appears that there are two choices involved in order to represent a tetrahedron: the choice of the canonical tetrahedron, and the choice of vertex ordering. This raises an important question: Does the interpolation path vary if we choose a different canonical tetrahedron and ordering of vertices in a tetrahedron? These questions are addressed in Section 5, where we discuss the implications of the choices made.

3.1 Matrix Representation of Transformations

Since the individual transformations differ in the number of parameters, these need to be embedded in appropriate matrices before they can be composed to obtain a 3D affine transformation.

The rigid transformation $E \in SE(3)$, consisting of a rotation R and translation vector d, will be represented as a 4×4 matrix: $E = \begin{bmatrix} R & d \\ 0_3^T & 1 \end{bmatrix}$. The uniform scaling $s \in \mathbb{R}^+$ will be represented by the matrix $S = \begin{bmatrix} sI_3 & 0_3 \\ 0_3^T & 1 \end{bmatrix}$, while the 3×3 shear transformation matrices will be embedded in the upper left submatrix of I_4, i.e. $\begin{bmatrix} A & 0_3 \\ 0_3^T & 1 \end{bmatrix}$. We will denote both the 3×3 and 4×4 shear matrices with the same notation, typically A. The 3D affine transformation that deforms a given tetrahedron into the canonical tetrahedron is given by,

$$\mathbb{T} = ASE. \tag{1}$$

The transformation $\mathbb{T}$ is represented by the triplet $(E, A, s) \in \mathcal{M} := SE(3) \times G_A \times G_S$, and thus, any tetrahedron can be represented by an element of the set $\mathcal{M}$. Note that each of the three sets of transformations used above forms a Lie group[4], and thus the direct product $\mathcal{M}$ is also a Lie group. We give a short introduction to Lie groups below, and refer the reader to [Gallier and Quaintance 2017; Postnikov 2001] for more details.

[2] A tetrahedron is *non-degenerate* if the volume enclosed by the tetrahedron is strictly positive.

[3] Two tetrahedrons with vertices $p_i \in \mathbb{R}^3, i = 0, \ldots, 3$ and $q_i \in \mathbb{R}^3, i = 0, \ldots, 3$ listed in this order, are said to have same *orientation* if $sign(det(P)) = sign(det(Q))$, where $P = [\tilde{p}_0\ \tilde{p}_1\ \tilde{p}_2\ \tilde{p}_3] \in \mathbb{R}^{4\times4}$, $Q = [\tilde{q}_0\ \tilde{q}_1\ \tilde{q}_2\ \tilde{q}_3] \in \mathbb{R}^{4\times4}$, and $\tilde{p_i} \in \mathbb{R}^4$ is the homogeneous coordinate representation of $p_i \in \mathbb{R}^3$.

[4] $SE(3)$ and G_S are well known Lie groups, while the proof that G_A is a Lie group is given in Appendix A.1.

3.2 Introduction to Matrix Lie Groups

Lie groups lie at the intersection of groups and manifolds, i.e, a Lie group $\mathcal{G}$ is a group $(\mathcal{G}, \cdot)$ and a smooth manifold, such that the group multiplication and inversion operations are smooth. We work with matrix Lie groups, which are closed subgroups of the Lie group $GL(n,\mathbb{R})$ (group of all $n \times n$ real-valued invertible matrices). Each element of any such closed subgroup of $GL(n,\mathbb{R})$ can be thought of as a point in $\mathbb{R}^{n^2}$ (for example, by concatenating columns or rows), and the set of all such points forms an embedded submanifold of $\mathbb{R}^{n^2}$. Let $\mathcal{G}$ be a matrix Lie group with dimension m as a manifold. The Lie algebra, denoted by $\mathfrak{g}$, is defined as $\mathfrak{g} = \{X \in M_n(\mathbb{R}) \mid \exp(X) \in \mathcal{G}\}$, where $M_n(\mathbb{R})$ denotes the set of all $n \times n$ real matrices, and exp denotes the matrix exponential: $\exp(X) = \sum_{i=0}^{\infty} \frac{X^i}{i!}$. Note that $\mathfrak{g}$ is an m-dimensional vector space. It can be shown that the tangent space at identity $e \in \mathcal{G}$, denoted by $T_e\mathcal{G}$ can be identified with the Lie algebra. Thus one can think of $\mathfrak{g}$ as a linearized approximation of $\mathcal{G}$ around the point e. The tangent space at an arbitrary point $p \in \mathcal{G}$ is defined using left (or right) translation as $T_p\mathcal{G} = \{pX \mid X \in \mathfrak{g}\}$.

The Lie algebraic exponential map is a smooth map between $\mathfrak{g}$ and $\mathcal{G}$, such that it maps an element $v \in \mathfrak{g}$ to the element $\gamma(1) = \exp(v) \in \mathcal{G}$, where $\gamma : \mathbb{R} \to \mathcal{G}$ is the unique integral curve[5] of the left-invariant vector field generated by v. In the matrix Lie group case, the Lie algebraic exponential and log maps coincide with the matrix exponential and log map (which is the inverse of the exponential wherever possible), and can be used to go back and forth between $\mathcal{G}$ and $\mathfrak{g}$, $\exp : \mathfrak{g} \to \mathcal{G}$ and $\log : \mathcal{G} \to \mathfrak{g}$. In what follows, both Lie algebraic and matrix exponential and log maps will be referred to simply as exponential and log maps.

A locally length-minimizing curve, or *geodesic*, between e and $p \in \mathcal{G}$ is given by $\exp(t\log(p))$, with the property that the geodesic $c(t)$ between two points $p, q \in \mathcal{G}$ can be obtained by first translating both p and q on the left by p^{-1} (thus getting points e and $p^{-1}q$), computing the geodesic between the resulting points, and then left translating this new geodesic back by p, i.e., $c(t) = p\exp(t\log(p^{-1}q))$, (cf. Figure 2).

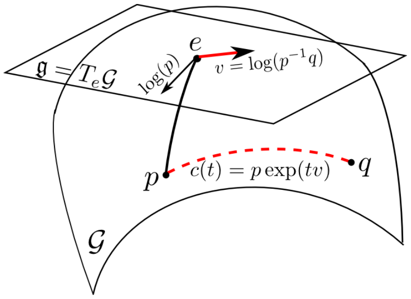

Fig. 2. Exponential & Log map on a matrix Lie group $\mathcal{G}$. The vector $v = \log(p^{-1}q) \in \mathfrak{g}$ shown in red, can be used to compute the geodesic between p and q as $c(t) = p\exp(tv)$, shown as a red curve with dashes.

3.3 Lie Group Representation of a Tetrahedron

The group operation on Lie group $\mathcal{M}$ is defined as: $(E_1, A_1, s_1) \cdot (E_2, A_2, s_2) = (E_1E_2, A_1A_2, s_1s_2)$. Henceforth the group operation $p \cdot q$, for $p, q \in \mathcal{M}$ will simply be denoted by pq. The Lie algebra $\mathfrak{m}$ of $\mathcal{M}$ is defined as $\mathfrak{m} = \mathfrak{se}(3) \times \mathfrak{g}_A \times \mathfrak{g}_S$, where

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} \Omega & d \\ 0_3^T & 0 \end{bmatrix} \in M_4(\mathbb{R}) \mid \Omega \in M_3(\mathbb{R}), d \in \mathbb{R}^3, \Omega = -\Omega^T \right\},$$

$$\mathfrak{g}_A = \left\{ \begin{bmatrix} 0 & \bar{\alpha}_1 & \bar{\alpha}_3 \\ 0 & \bar{\alpha}_2 & \bar{\alpha}_4 \\ 0 & 0 & \bar{\alpha}_5 \end{bmatrix} \mid \bar{\alpha}_i \in \mathbb{R}, i = 1, \ldots, 5 \right\},$$ and $\mathfrak{g}_S = \mathbb{R}$, are the Lie algebras of the Lie groups $SE(3)$, G_A and G_S, respectively. Thus $\mathcal{M}$ is a 12 dimensional Lie group (6 for $SE(3)$, 5 for G_A and 1 for G_S). The exponential and logarithm map for $\mathcal{M}$ is obtained by concatenating the exponential and logarithm map for each of three components: $SE(3)$, G_A and G_S, i.e., $\exp : \mathfrak{m} \to \mathcal{M} = (\exp : \mathfrak{se}(3) \to SE(3), \exp : \mathfrak{g}_A \to G_A, \exp : \mathfrak{g}_S \to G_S)$ and $\log : \mathcal{M} \to \mathfrak{m} = (\log : SE(3) \to \mathfrak{se}(3), \log : G_A \to \mathfrak{g}_A, \log : G_S \to \mathfrak{g}_S)$. The exponential and logarithm maps for the groups $SE(3)$ and G_S are known in closed form [Freifeld and Black 2012; Gallier and Quaintance 2017] and are listed in the Appendix, while the corresponding derivations for Lie group G_A is given in the accompanying supplementary material.

Thus, a non-degenerate tetrahedron can be represented using an appropriate element of $\mathcal{M}$, with respect to a chosen canonical tetrahedron. Given an orientation-preserving affine transformation, we can apply the inverse of it on a chosen canonical tetrahedron, to obtain another tetrahedron, whose Lie group representation can be used to represent the given affine transformation. The existence and uniqueness of our representation for a tetrahedron, or for an orientation-preserving affine transformation rely on the following two results. (1) Given a pair of oriented tetrahedrons with correspondence between vertices, a unique orientation-preserving affine transformation exists transforming one tetrahedron into the other. (2) Additionally, given a fixed ordering of vertices of the tetrahedron, a unique decomposition in the proposed representation exists. These two properties allow us to represent any tetrahedron, or any orientation-preserving affine transformation, uniquely with the proposed framework. These results are discussed in the following subsection.

3.4 Existence and Uniqueness

We will represent a tetrahedron in $\mathbb{R}^3$ via its vertices (u_0, u_1, u_2, u_3), and we will collect the coordinates of the vertices as column vectors in a matrix. The ordering of vertices is assumed to be fixed, and with a little abuse of notation, we will use the same symbol to denote a tetrahedron, and the matrix (of size 4×4) containing its homogeneous vertex coordinates, wherever required.

Theorem 1. *Let Δ_X and Δ_Y represent two tetrahedrons in $\mathbb{R}^3$ with correspondence between vertices given, and vertices placed in a fixed order. Let $\mathbb{T}$ denote an invertible orientation-preserving affine transformation. Then the following is true:*

(1) *For any pair of non-degenerate tetrahedrons Δ_X and Δ_Y with the same orientation, a unique orientation-preserving affine transformation $\mathbb{T}$ exists such that $\mathbb{T}\Delta_X = \Delta_Y$.*
(2) *For a given orientation-preserving affine transformation $\mathbb{T}$, a pair of non-degenerate tetrahedrons Δ_X and Δ_Y exist such*

[5] An integral curve of a vector field is a curve whose tangent vector at any point of the curve is equal to the given vector field at the corresponding point.

that $\mathbb{T}\Delta_X = \Delta_Y$. If the transformation $\mathbb{T}$ and one of the two tetrahedrons are given, the other tetrahedron can be computed uniquely.

(3) *Given an orientation-preserving affine transformation $\mathbb{T}$ between tetrahedrons Δ_X and Δ_Y, $\mathbb{T}$ can be uniquely decomposed into components (E, A, S) such that $\mathbb{T} = ASE$.*

Proof. Refer to Appendix A.1. □

Part (1) and (2) of Theorem 1 together show that there exists a bijection between the set of orientation-preserving affine transformations and the set of pairs of non-degenerate tetrahedrons (with one of the tetrahedrons fixed). With the decomposition given by part (3) of Theorem 1, a unique decomposition (corresponding to a fixed ordering of vertices) of the transformation $\mathbb{T}$ is possible.

4 INTERPOLATION ALGORITHM & PROPERTIES

Given a 3D orientation-preserving affine transformation $\mathbb{T}$, let $p = (E, A, s) \in \mathcal{M}$ be its proposed Lie group representation. We propose to interpolate each of these three components on their respective Lie groups using appropriate exponential and logarithm maps, and combine the resultant components to obtain an interpolation of the given affine transformation. Note that the exponential and logarithm maps are known in closed form for each of the three components, but not for the group of general orientation-preserving affine transformations. Thus, the interpolated affine transformation is given as

$$\mathbb{T}_t = \exp(t \log A) \exp(t \log S) \exp(t \log E), \tag{2}$$

where $\log A$, $\log S$ and $\log E$ are logarithms of the matrices A, S and E respectively. The Lie group representation of the interpolated transformation between the identity I and transformation $\mathbb{T}$ at time $t \in [0, 1]$, is given by

$$p(t) = \exp(t \log p). \tag{3}$$

Note that for A and S, the exponential and log maps are considered in the form of matrices of compatible size. Closed-form expressions of these maps are given in Appendix A.2. Similarly, let Δ_p, Δ_q be two non-degenerate tetrahedrons in $\mathbb{R}^3$ with given vertex correspondence and vertex order to be interpolated, and let Δ represent the canonical tetrahedron. Let $p, q \in \mathcal{M}$ be the Lie group elements representing the tetrahedrons Δ_p, Δ_q respectively. The representation of the interpolated tetrahedron between Δ_p and Δ_q is given by

$$p \exp(t \log p^{-1} q), \quad t \in [0, 1] \tag{4}$$

In [Rossignac and Vinacua 2011], the interpolation of an affine transformation $\mathbb{T}$ is defined to be steady if $\mathbb{T}_t = \mathbb{T}^t, \quad t \in \mathbb{R}$. The following result specifies conditions under which our interpolation is steady, whose proof is given in Appendix A.1.

Lemma 1. *(Steady interpolation) If the given transformation $\mathbb{T}$ consists either of only one of the three components: A, S or E, or a combination of A and S, then the proposed interpolation is steady.*

The interpolation scheme proposed in [Rossignac and Vinacua 2011] is always steady, but does not exist in cases of large rotation and shear. Our solution, although not always steady, always exists. One can thus think of our scheme as a decomposition of the given affine transformation into the specified components, followed by a steady interpolation in each of the components, concluding with a composition of the components.

In the following lemmas, we provide a set of properties preserved by our interpolation algorithm. The proofs are given in Appendix A.1. These properties are essential in order to get intuitive interpolation paths.

Lemma 2. *(Volume Preservation) If the given transformation $\mathbb{T}$ is volume-preserving, i.e.,* $\det(\mathbb{T}) = 1$, *then the interpolated transformations $\mathbb{T}_t, 0 \leq t \leq 1$ defined in Equation* (2) *are also volume-preserving.*

Lemma 3. *(Monotonic variation of volume) If the given transformation $\mathbb{T}$ increases (decreases) the volume, i.e.,* $\det(\mathbb{T}) > 1$ $(\det(\mathbb{T}) < 1)$, *then the interpolated transformations $\mathbb{T}_t, 0 \leq t \leq 1$ monotonically increase (decrease) the volume.*

Lemma 4. *(Isometric Transformations) Assuming that the transformation $\mathbb{T}$ is isometric, i.e.,* $||\mathbb{T}(x - y)|| = ||x - y||, \forall x, y \in \mathbb{R}^3$, *all the interpolated transformations $\mathbb{T}_t, 0 \leq t \leq 1$ are also isometric.*

Lemma 5. *(Reversibility) Let Δ_p and Δ_q be two tetrahedrons. Let $\Delta_p^q(t), t \in [0, 1]$ represent the interpolated tetrahedron at time t considering tetrahedron Δ_p as source, and tetrahedron Δ_q as target. Then* $\Delta_p^q(t) = \Delta_q^p(1 - t), \forall t \in [0, 1]$.

Intuitively, this rests on the fact that geodesics are reversible, which is the key construct of our interpolation algorithm.

4.1 Smooth Interpolation of Multiple 3D Affine Transformations

It is often required to interpolate a sequence of affine transformations $\mathbb{T}_i, i = 0, \dots, n$ smoothly. Using the interpolation algorithm given in the previous section for each consecutive pair of transformations from the given sequence will produce a continuous but not differentiable interpolation. Lack of smoothness produces less appealing interpolations.

We summarize the algorithm proposed in [Li and Hao 2006] in order to obtain a C^1 interpolation, and refer the reader to this paper for higher-order smoothness. Let the decomposition of each $\mathbb{T}_i$ be given by $\mathbb{T}_i = A_i S_i E_i, i = 0, \dots, n$. We first compute a C^1 interpolation of individual components in the respective Lie group. We will present the algorithm for a smooth (C^1) interpolation in the Lie group G_A; the algorithm for interpolation in the other Lie groups remain the same.

Let $N_i^A(t), t \in [0, 1], i = 0, \dots, n-1$ denote the i^{th} segment of the C^1 interpolating curve in G_A. The constraints that ensure a C^1 interpolation are: $N_i^A(1) = N_{i+1}^A(0) = A_{i+1}, i = 0, \dots, n-2, N_0^A(0) = A_0, N_{n-1}^A(1) = A_n$, and $\frac{d}{dt} N_i^A(1) = \frac{d}{dt} N_{i+1}^A(0), i = 0, \dots, n-2$. In order to ensure that a solution to these constraints exists, we define the curve as a quadratic curve in terms of two free variables $B_i, C_i \in \mathfrak{g}_A, i = 1, \dots, n-1$, as $N_i^A(t) = A_i \exp(t B_i) \exp(t^2 C_i), i = 0, \dots, n-1$. The constraints on the curve yield a set of equations that the free variables must satisfy:

$$C_i = \log\left(\exp(-B_i) A_i^{-1} A_{i+1}\right), i = 0, \dots, n-1, \tag{5}$$

$$B_{i+1} = A_{i+1}^{-1} A_i B_i A_i^{-1} A_{i+1} + 2C_i, i = 0, \dots, n-2, \tag{6}$$

while $B_0 \in \mathfrak{g}_A$ is a free variable. In our experiments we set $B_0 = \log(A_0^{-1}A_1)$. Using appropriate exponential and logarithm maps, one can similarly determine C^1 curves $N_i^S(t)$, $N_i^{SE(3)}(t)$ in the Lie groups G_S and $SE(3)$ respectively. The C^1 affine transformation curve is then obtained by composing these individual curves: $\mathbb{T}_i(t) = N_i^{SE(3)}(t) \cdot N_i^S(t) \cdot N_i^A(t), i = 0, \ldots, n-1, t \in [0,1]$.

One can also use a subdivision scheme on a curve interpolating the given n transformations. The given curve can be first sampled to obtain mn transformations with $m > 1$, followed by the interpolation procedure described above to produce a smoother interpolation. This is demonstrated via experiments in Section 6.

In the next section, we discuss the invariance properties of the proposed interpolation scheme with respect to the choice of the canonical tetrahedron and choice of vertex ordering.

5 CHOICE OF CANONICAL TETRAHEDRON AND RE-ORDERING

We first show that the proposed approach is invariant to the choice of canonical tetrahedron, followed by an investigation of the effects of change in vertex ordering.

5.1 Effect of changing the canonical tetrahedron

In the proposed Lie group representation of a tetrahedron, the shear component assumes that the tetrahedron is aligned in a particular manner by the rigid component. In case a different canonical tetrahedron is chosen, the assumption on alignment may be violated. Rather than changing the structure of the shear matrix, we use the same rigid component used with the original canonical tetrahedron in the representation with respect to the new canonical tetrahedron. Thus, for a tetrahedron Δ_p, if the representation in terms of the usual canonical tetrahedron is (E_p, A_p, s_p), then its representation with respect to a new canonical tetrahedron Δ_1 is (E_p, A_{p_1}, s_{p_1}), while the actual affine transformation between Δ_p and Δ_1 is

$$\mathbb{T}_1 = E_{\Delta_1}^{-1} A_{p_1} S_{p_1} E_p, \tag{7}$$

where E_{Δ_1} is the rigid component in the representation of Δ_1 with respect to Δ.

Theorem 2. *(Invariance to Canonical tetrahedron): Let Δ_1 and Δ_2 be two arbitrary canonical tetrahedrons, and Δ_p and Δ_q be two tetrahedrons to be interpolated. Let (E_p, A_{p_i}, s_{p_i}) and $(E_q, A_{q_i}, s_{q_i}), i = 1, 2$ denote the Lie group representations of tetrahedrons Δ_p and Δ_q with respect to canonical tetrahedrons $\Delta_i, i = 1, 2$ respectively. Let $(E_{c_i}, A_{c_i}, s_{c_i}), i = 1, 2$ be the interpolated Lie group elements obtained using canonical tetrahedrons $\Delta_i, i = 1, 2$, respectively. Then, $(E_{c_i}, A_{c_i}, s_{c_i}), i = 1, 2$ represent the same tetrahedron.*

Proof. Refer to Appendix A.1. □

Intuitively, this is analogous to interpolation between two points in an n-dimensional Euclidean space $\mathbb{E}^n$. Once an origin is fixed, the points can be represented as vectors in $\mathbb{R}^n$ and interpolation can be carried out using the familiar convex combination of the vectors representing the points. With a different choice of origin, the representation of the two points and the interpolated point will change, but the points themselves remain the same. In our case, we interpolate between points of a manifold instead of a vector space, using a generalization of the convex combination operation in terms of the exponential and log maps. In effect, what we show is that the choice of canonical tetrahedron is analogous to the choice of origin in a Euclidean space, and therefore it does not affect the interpolation process.

5.2 Effect of re-ordering

The Lie group representation of a tetrahedron in the proposed framework depends on the choice of the first face and edge (ordering of vertices in the face-list), since the rigid transformation is chosen to align the first face and edge. Let $O_i, i = 1, 2$ represent two distinct orderings of vertices of any tetrahedron. Let $p_{O_i}, i = 1, 2$ and $q_{O_i}, i = 1, 2$ denote the Lie group elements representing the tetrahedrons Δ_p and Δ_q, in the vertex orderings $O_i, i = 1, 2$, respectively. Let $V_{O_i}^p, i = 1, 2$ and $V_{O_i}^q, i = 1, 2$ denote the 4×4 matrices of vertices of the tetrahedrons Δ_p and Δ_q in homogeneous coordinates in column order $O_i,\ i = 1, 2$.

5.2.1 Isometric transformations & uniform scaling. Under the assumption that tetrahedrons Δ_p and Δ_q exhibit only rigid transformation and/or uniform scaling, there exists an $G_S \times SE(3)$ element, say (s, E) such that $V_{O_i}^q = SEV_{O_i}^p, i = 1, 2$. Thus the Lie group elements of the source and target tetrahedrons under different orderings are related by $q_{O_i} = p_{O_i}(E^{-1}, I_3, \frac{1}{s}), i = 1, 2$. The tangent vectors for both orderings is then $\log((p_{O_i})^{-1} q_{O_i}) = \log((E^{-1}, I_3, \frac{1}{s})), i = 1, 2$. Hence, the interpolation process remains invariant to re-ordering if the underlying transformation is an isometry and/or uniform scaling.

5.2.2 Non-isometric transformations. In cases where the source and target tetrahedrons exhibit shear transformation other than rigid and uniform scaling, the interpolation path depends on the choice of the first face and edge. Let $\mathbb{T}_t^{O_1}$ and $\mathbb{T}_t^{O_2}$ be the interpolated transformations at time $t \in [0,1]$ corresponding to two different orderings of the tetrahedron vertices given by

$$\mathbb{T}_t^{O_i} = \exp(t \log A_{O_i}) \exp(t \log S_{O_i}) \exp(t \log E_{O_i}), i = 1, 2,$$

where $(E_{O_i}, A_{O_i}, s_{O_i})$ are the corresponding Lie group representations, and we have assumed the canonical tetrahedron from Section 3. The change in the interpolated path can be represented by the change in the corresponding transformations i.e. $u_{O_1,O_2} = \max_{t\in[0,1]} ||\mathbb{T}_t^{O_1} - \mathbb{T}_t^{O_2}||_F$. We use the maximum and median of u over all orderings in order to capture the amount of change in interpolation paths under vertex re-orderings. In Figure 9, we empirically demonstrate that the change in path depends on the amount of shear component. A theoretical analysis of dependence of interpolation path on vertex ordering in the presence of shear needs to be carried out, so that users can make an informed choice of a particular vertex ordering.

We now provide details of the experiments and their results using the proposed framework and a comparison with other state-of-the-art approaches.

6 RESULTS

In what follows, we will refer to the interpolation approach in [Rossignac and Vinacua 2011] as SAM (Steady Affine Morph). Once we have the interpolated affine transformations, the results can be shown on different 3D objects. We present results using tetrahedrons and cuboids. We also include MATLAB figures and source codes in the accompanying supplementary material.

6.1 Comparison with the state-of-the-art

We begin with focusing on planar affine transformations. Such an affine transformation can be used as a map between triangles in $\mathbb{R}^2$. In Figure 3, we show results of planar interpolation on several cases for the following existing methods of interpolation: Shoemake1992 & Alexa2000 ([Shoemake 1992] [Alexa et al. 2000]), Alexa2002 [Alexa 2002], Sumner2005 [Sumner et al. 2005] and SAM [Rossignac and Vinacua 2011]. Translation is present in all examples. In addition, examples 2 and 4 have only rotation and shear component respectively, while examples 1, 3 and 5 have a varying degree of rotation and shear components. It can be seen that the results produced by the proposed approach and SAM produce very similar and intuitive results. Also note that example 5 consists of a large shear and rotation transformation, in which case the SAM interpolation approach fails, while the proposed approach is able to produce an intuitive interpolation.

Moving onto 3D affine transformations, in Figure 4, we produce interpolations for affine transformations with constant rotation about the z-axis, and gradually increasing degree(top to bottom, left to right) of shear, using SAM and the proposed approach. When the shear component becomes large, SAM fails to produce an interpolation (Row 4, Column 3 in Figure 4). One can also observe in the bottom row of Figure 4 that the trajectory corresponding to SAM changes considerably before it fails to produce an interpolation, while the proposed method does not alter drastically, and produces an interpolation for all cases.

The Average Relative Acceleration (ARA) measure is used in SAM, to quantify the steadiness of the interpolation path. It is defined in [Rossignac and Vinacua 2011] as: *the average magnitude (over space and time) of the acceleration vector by which the relative velocity expressed in the moving frame changes over time.* The ideal value of ARA is zero, and by construction, SAM achieves an ARA of 0. Since our algorithm does not produce steady interpolation in cases other than those mentioned in Lemma 1, we include an empirical comparison of the ARA measure of the proposed approach with SAM for the examples from Figure 3, in Table 1. The step size for the grid and the number of intermediate poses used in computing the ARA are .01 and 50, respectively. Although the proposed approach is not steady in all cases, the ARA measures are at most a magnitude off from those obtained by SAM for the corresponding discretization.

Table 1. Comparison of ARA measure proposed in [Rossignac and Vinacua 2011], with the proposed approach for examples from Figure 3. ARA is computed by taking a grid sampled at an interval of .01 and for 50 intermediate poses.

Approaches	SAM	Proposed
Example 1	9.0282e-05	1.1218e-04
Example 2	2.7515e-06	2.3321e-06
Example 3	3.1247e-06	1.3392e-05
Example 4	6.9723e-05	6.9486e-05
Example 5	-	3.8634e-05

6.2 Extrapolation

Rather than restricting the computation of $\mathbb{T}_t$ for $t \in [0, 1]$, we can extrapolate the transformation by computing $\mathbb{T}_t$ for t beyond this range. In Figure 5, we show results of extrapolating the given 3D affine transformation using our approach and SAM. The two approaches produce the same result in case the transformation is a combination of shear and translation (refer to Figure 5, center column), while the translation component differs in case the transformation consists of rotation, scaling and translation (refer to Figure 5, left). This is evident from the last row in Figure 5, where we plot the edge lengths of one triangle (base) and the distance of the fourth vertex from the base triangle for each tetrahedron obtained. The edge lengths for tetrahedrons produced by SAM are shown in red, while those using our approach are shown in yellow. We emphasize the difference in the two approaches in case the transformation consists of rotation, shear, and translation. As can be seen in Figure 5 (right), a regular tetrahedron is rotated, translated and sheared to make it non-regular. When our approach is used to extrapolate this transformation, it yields more irregular tetrahedrons, while SAM reproduces a regular tetrahedron at some time $t > 1$, as can be observed from plots of lengths of tetrahedron sides in Figure 5 (bottom row, right). The lengths in the first two examples of each tetrahedron are same for both approaches (Figure 5, bottom row: left, center).

6.3 Smooth interpolation

In Figure 6, smooth variations in the interpolation are shown. Interpolations and extrapolations between two pair of 3D affine transformations (shown in blue) are computed (shown in green). Each pair of interpolated/extrapolated transformations are then interpolated/extrapolated. The time parameter values used in all cases are $t = -.2, .2, .4, .6, .8, 1.2$. The result shows that the interpolated transformation varies smoothly in each direction. If, instead of generating interpolations row-wise and then column-wise, we first interpolate column-wise and then row-wise to generate the grid, the results only differ slightly. A theoretical analysis needs to be performed to analyze this behaviour.

6.4 Interpolating multiple affine transformations

We demonstrate our algorithm to interpolate multiple affine transformations as discussed in Section 4.1. In Figure 7, we show two examples of a smooth interpolation between multiple 3D affine transformations containing scale, rigid and shear components. In order to demonstrate the subdivision scheme, we begin with continuous (but not necessarily differentiable) curves that interpolate the given affine transformations as shown in the left column of Figure 8. By adding transformations obtained at $t = 0.5$ to the set of given

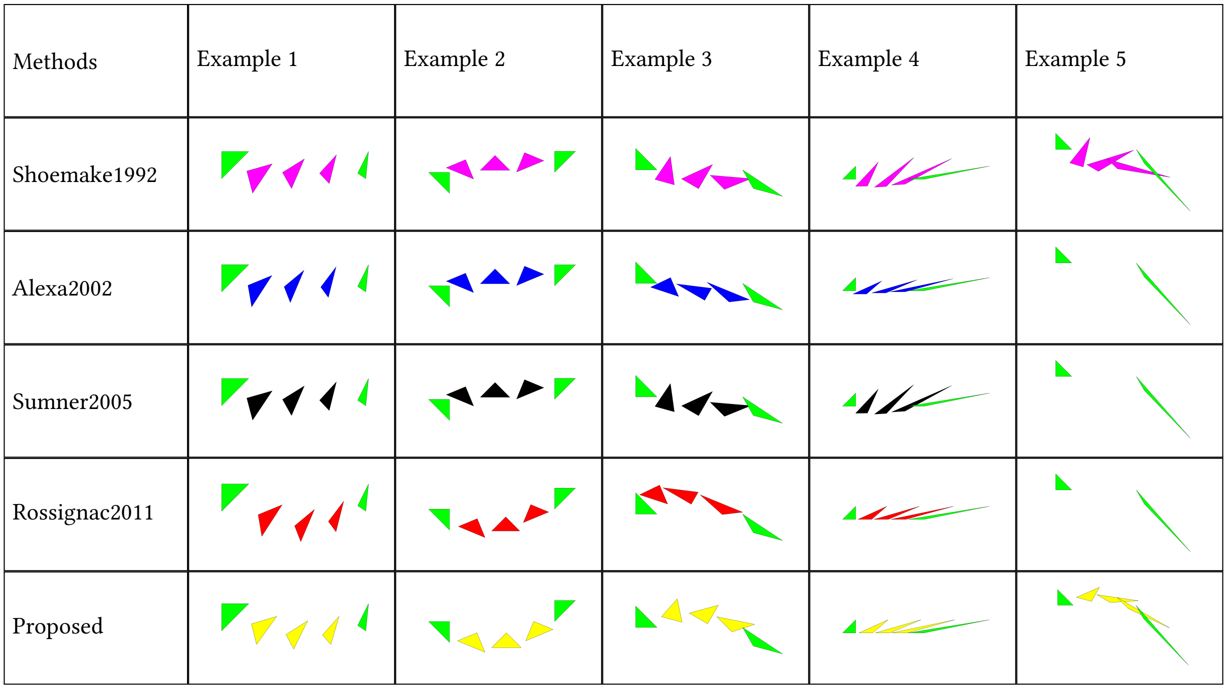

Fig. 3. Examples of planar interpolation between triangles in $\mathbb{R}^3$. Row-wise (top-down) Interpolation results using method proposed in Shoemake1992 & Alexa2000 ([Shoemake 1992] [Alexa et al. 2000]), Alexa2002 [Alexa 2002], Sumner2005 [Sumner et al. 2005], SAM [Rossignac and Vinacua 2011] and proposed approach, for 5 different examples (column-wise). Triangles in green denote the source and target triangles. Interpolated results are produced for $t = 0.25, 0.5, 0.75$.

affine transformations followed by a smooth interpolation yields a smoother trajectory as shown in the right column of Figure 8.

6.5 Effect of reordering

To demonstrate the effect of reordering, we provide a comparison of the interpolation paths obtained with different vertex orderings as discussed in Section 5 by taking examples under two scenarios. In both scenarios, we use a regular tetrahedron inscribed in a unit sphere as the source tetrahedron. In the first case, the target tetrahedron is obtained by keeping two vertices of the source tetrahedron fixed and moving the third and fourth vertex along circles on orthogonal planes, lying on the unit sphere. For the second case, the third and the fourth vertex of the source tetrahedron are moved along the lines given by $x = constant$ and $y = constant$ respectively. Row 1 in Figure 9 demonstrates these deformations. The actual difference in any two orderings, say O_1 and O_2 is captured by $\max_{t\in[0,1]} ||\mathbb{T}_t^{O_1} - \mathbb{T}_t^{O_2}||_F$. Since there are 12 different orderings for a tetrahedron, we plot the maximum (Row 2) and median (Row 3) of differences of interpolation paths defined above, for each deformation, over all orderings in Figure 9. As can be observed from the figure, as the shear component increases (moving away from center of plot), the maximum and median deviation in path difference also increases. Actual interpolation paths for two instances of both the cases are included in Figures 10 (Case 1) and 11 (Case 2). It again points to the fact that the higher the shear component in the affine transformation, the higher the difference in the paths obtained due to different re-orderings.

7 CONCLUSION AND DISCUSSION

In this work, we propose a framework to interpolate an orientation-preserving affine transformation, based on a Lie group representation of the transformation. The proposed framework always provides a solution, contrary to some prior work, while preserving several important properties of the original transformation, like isometry and volume preservation. The approach is invariant to the choice of canonical tetrahedron, while the degree to which vertex ordering affects the interpolation has been analyzed in detail.

Volumetric mesh deformation and interpolation methods can be developed based on our interpolation method. Our method can be applied on each pair of constituent tetrahedrons in case of mesh interpolation. This will in general not guarantee a valid volumetric mesh. An appropriate stitching process that binds individual tetrahedrons to form a valid volumetric mesh needs to be worked out. Similarly, for interactive volumetric mesh deformation, a blending process has to be developed in order to appropriately transform all tetrahedrons of the mesh.

REFERENCES

Marc Alexa. 2002. Linear combination of transformations. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 380–387.

Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and*

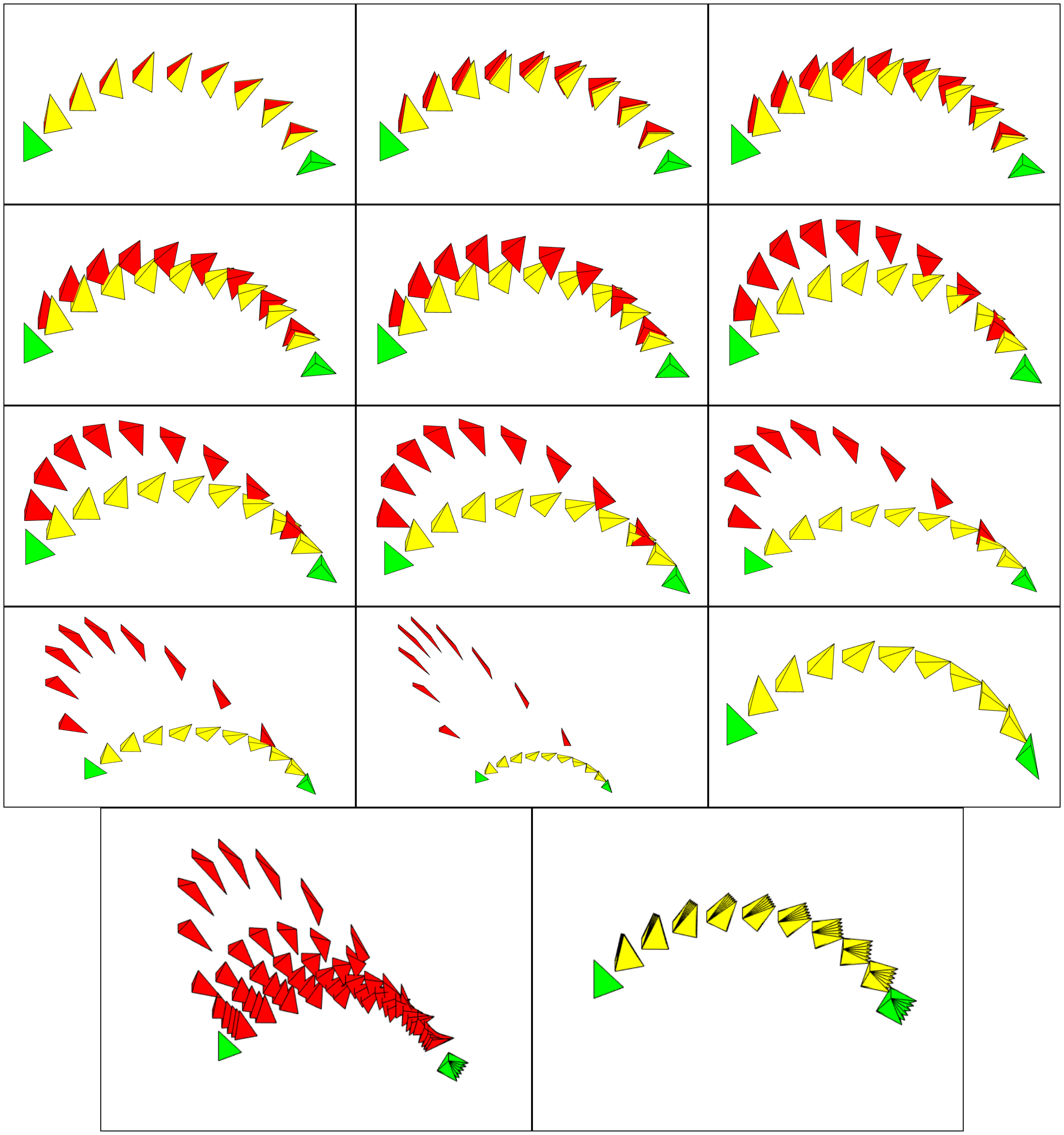

Fig. 4. A comparison between SAM and the proposed approach. An example with a constant rotation about the z axis, while the shear (of x coordinate by y coordinate) component is gradually increased from top to bottom in Rows 1 to 4, and left to right. While the source and target tetrahedrons are shown in green, the interpolated tetrahedrons produced by SAM are shown in red and those produced by the proposed algorithm are shown in yellow. As expected, when the shear component becomes large, the SAM interpolation does not exist (Row 4, right). (Bottom row): 6 of the above 12 interpolations for SAM and proposed approach are superimposed in order to compare the difference in the obtained interpolations as the shear component increases. The trajectory corresponding to SAM changes considerably due to the steadiness requirement before it fails to produce an interpolation, while the proposed method does not produce significantly different interpolations.

interactive techniques. ACM Press/Addison-Wesley Publishing Co., 157–164.

Kevin G Der, Robert W Sumner, and Jovan Popović. 2006. Inverse kinematics for reduced deformable models. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 1174–1179.

Oren Freifeld and Michael J. Black. 2012. Lie Bodies: A Manifold Representation of 3D Human Shape. In *European Conf. on Computer Vision (ECCV) (Part I, LNCS 7572)*.

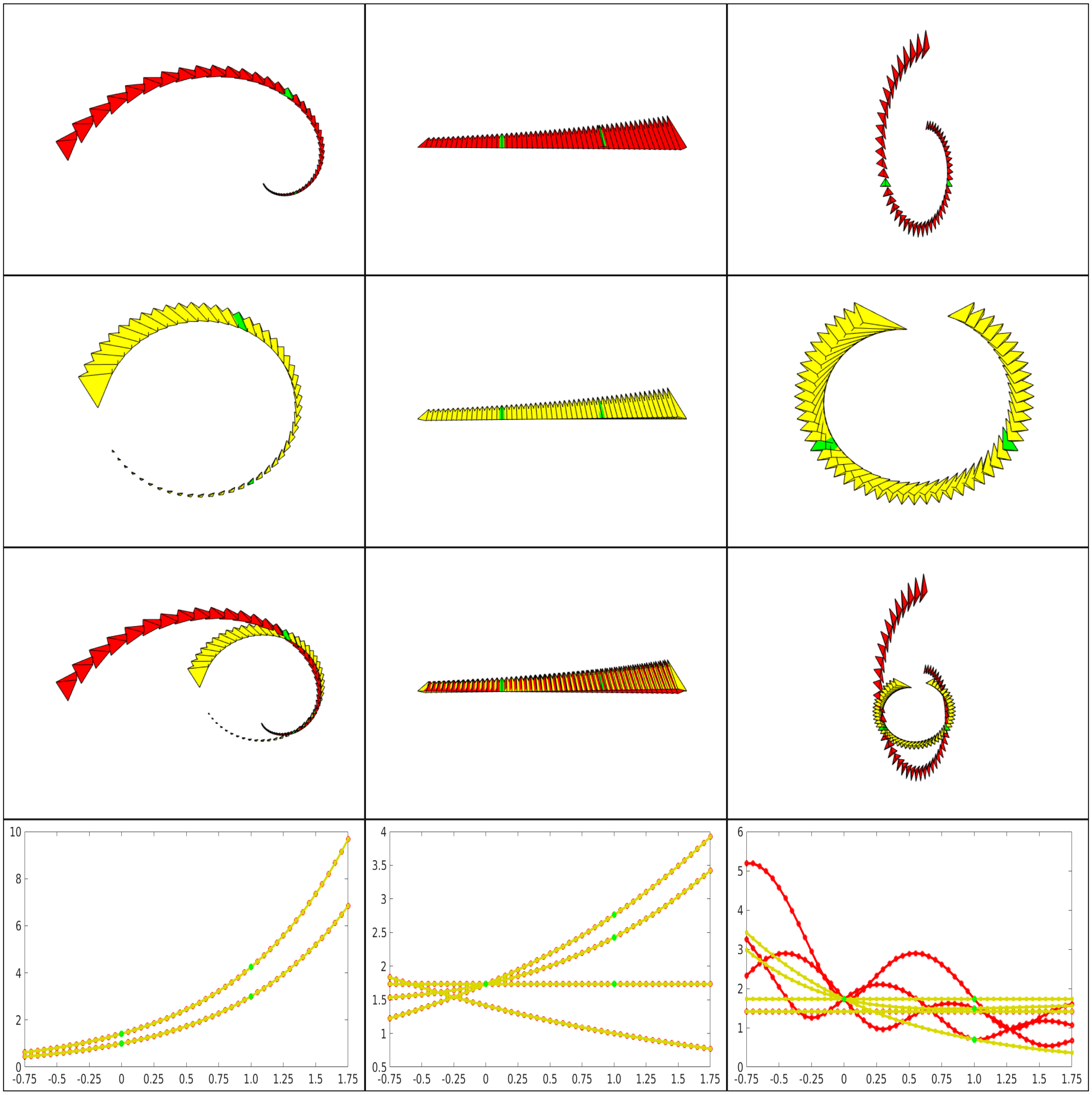

Fig. 5. Extrapolation and interpolation. Source and target tetrahedrons are in green. Results using SAM(Row 1), the proposed approach (Row 2), comparison of SAM and the proposed approach (Row 3) and edge lengths corresponding to the edges of the interpolated base triangles and distance of fourth vertex from the base triangle (Row 4: SAM in red, proposed approach in yellow). The affine transformation used is: Column 1- Rotation, scale and translation, Column 2- shear and translation, and Column 3- Rotation, shear and translation. Results are produced for $t \in [-0.75, 1.75]$ with a sampling interval of 0.05.

Springer-Verlag, 1–14.

Jean Gallier and Jocelyn Quaintance. 2017. Notes on Differential Geometry and Lie Groups. (2017). Available online at www.cis.upenn.edu/ jean/gbooks/manif.html.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. 2014. Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*.

S. Kaji, S. Hirose, S. Sakata, Y. Mizoguchi, and K. Anjyo. 2012. Mathematical Analysis on Affine Maps for 2D Shape Interpolation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*. Eurographics Association, Goslar Germany, Germany, 71–76. http://dl.acm.org/citation.cfm?id=2422356.2422368

Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)*

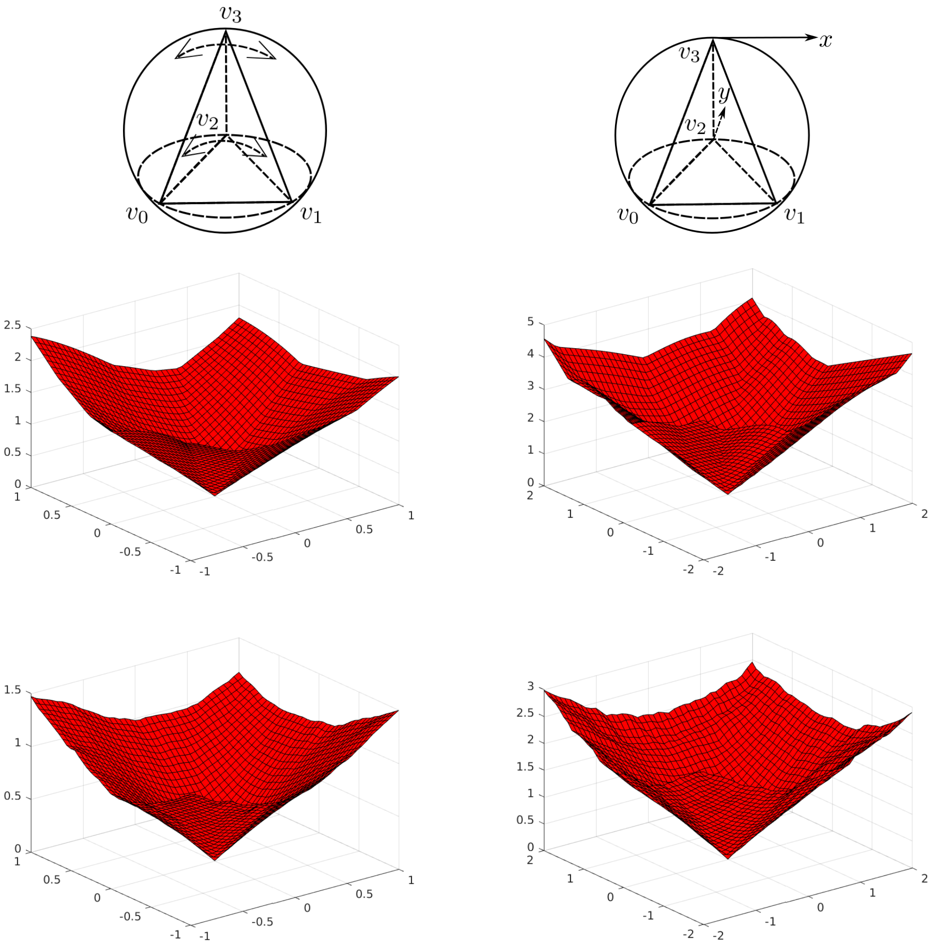

Fig. 9. (Row 1): The target tetrahedron is obtained by (left) rotating vertex v_2 along the circle on the sphere in the plane of vertices v_0, v_1, v_2, and (right) translating vertices v_2 and v_3 along the x and y axis respectively. These deformations were chosen to directly affect the shear component in the representation. (Row 2) Norm of maximum path change for each deformation over all vertex reorderings for the two cases of deformations. (Row 3) Norm of median path change for each deformation over all vertex reorderings for the two cases of deformations. The maximum and median, both increase with the increase in the shear component. The x-axis and y-axis represent the change in moving vertices for each case (angles in radian along the orthogonal circles in case 1).

27, 4 (2008), 105.

Ladislav Kavan and Jiří Žára. 2005. Spherical blend skinning: a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. ACM, 9–16.

Jun Li and Peng-wei Hao. 2006. Smooth interpolation on homogeneous matrix groups for computer animation. *Journal of Zhejiang University-SCIENCE A* 7, 7 (2006), 1168–1177. https://doi.org/10.1631/jzus.2006.A1168

Nadia Magnenat-Thalmann, Richard Laperrire, and Daniel Thalmann. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface*.

Richard M. Murray, S. Shankar Sastry, and Li Zexiang. 1994. *A Mathematical Introduction to Robotic Manipulation* (1st ed.). CRC Press, Inc., Boca Raton, FL, USA.

Hiroyuki Ochiai and Ken Anjyo. 2013. Mathematical description of motion and deformation: from basics to graphics applications. In *SIGGRAPH Asia 2013, Hong Kong, China, November 19-22, 2013, Courses*. 2:1–2:47. https://doi.org/10.1145/2542266.2542268

M. M. Postnikov. 2001. *Geometry VI: Riemannian Geometry*. Springer-Verlag Berlin Heidelberg.

Jarek Rossignac and Álvar Vinacua. 2011. Steady affine motions and morphs. *ACM Transactions on Graphics (TOG)* 30, 5 (2011), 116.

J. M. Selig. 2010. *Geometric Fundamentals of Robotics* (2nd ed.). Springer Publishing Company, Incorporated.

Ken Shoemake. 1985. Animating Rotation with Quaternion Curves. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 245–254. https://doi.org/10.1145/325165.325242

Ken. Shoemake. 1992. Matrix animation and polar decomposition. *Graphics Interface'92, 1992* (1992).

Robert W. Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. 2005. Mesh-based Inverse Kinematics. *ACM Trans. Graph.* 24, 3 (July 2005), 488–495. https://doi.org/10.1145/1073204.1073218

B. Whited, G. Noris, M. Simmons, R. Sumner, M. Gross, and J. Rossignac. 2010. BetweenIT: An Interactive Tool for Tight Inbetweening. *Comput. Graphics Forum (Proc. Eurographics)* 29, 2 (2010), 605–614.

Milos Zefran and Vijay Kumar. 1998. Interpolation schemes for rigid body motions. *Computer-Aided Design* 30, 3 (1998), 179 – 189. Motion Design and Kinematics.

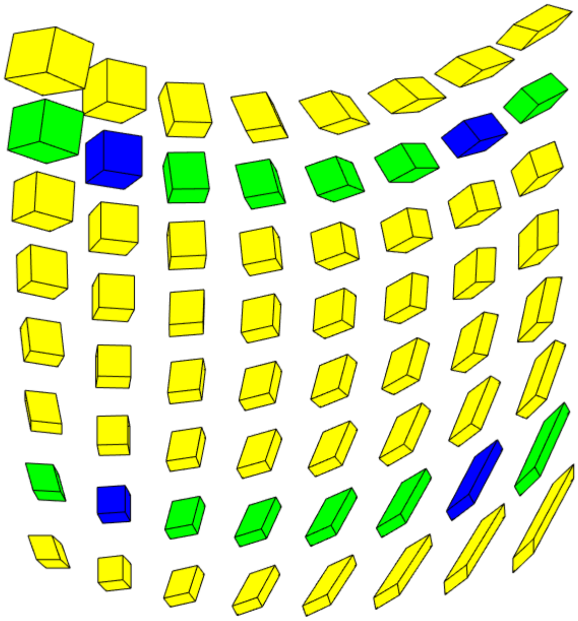

Fig. 6. Smooth interpolation using the proposed method is shown. Firstly, affine transformations between cuboids given in blue are interpolated/extrapolated to generate cubes shown in green for time $t = -0.2, 0.2, .4, .6, .8, 1.2$. Then affine transformations between corresponding green cubes are interpolated/extrapolated.

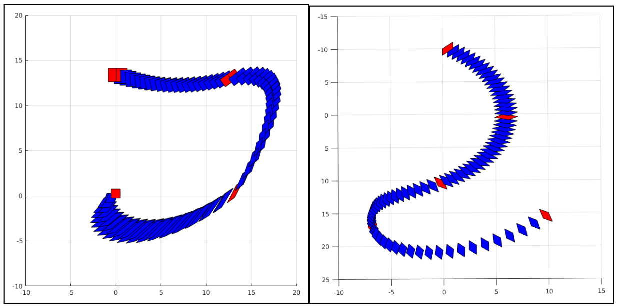

Fig. 7. Smooth Interpolation of multiple affine transformations. A sequence of affine transformation is provided (shown in red). Each pair of consecutive affine transformations is interpolated for a time step of $dt = .05$ (shown in blue).

A APPENDIX

A.1 Proofs of Theorems & Lemmas

A.1.1 G_A is a Lie group.

Proof. Using the fact that G_A is a subgroup in $GL(3)$ and Definition 4.6 in [Gallier and Quaintance 2017],

$$G_A = \left(\begin{bmatrix} 1 & \alpha_1 & \alpha_3 \\ 0 & \alpha_2 & \alpha_4 \\ 0 & 0 & \alpha_5 \end{bmatrix} | \alpha_2, \alpha_5 \in \mathbf{R}^+, \alpha_1, \alpha_3, \alpha_4 \in \mathbf{R} \right)$$

is linear Lie group as: $\forall A \in G_A, \phi : G_A \rightarrow \mathbf{R}^5$ given by $\phi(A) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) \in \mathbf{R}^5$, (ϕ, G_A) is a smooth manifold in $\mathbf{R}^5$. □

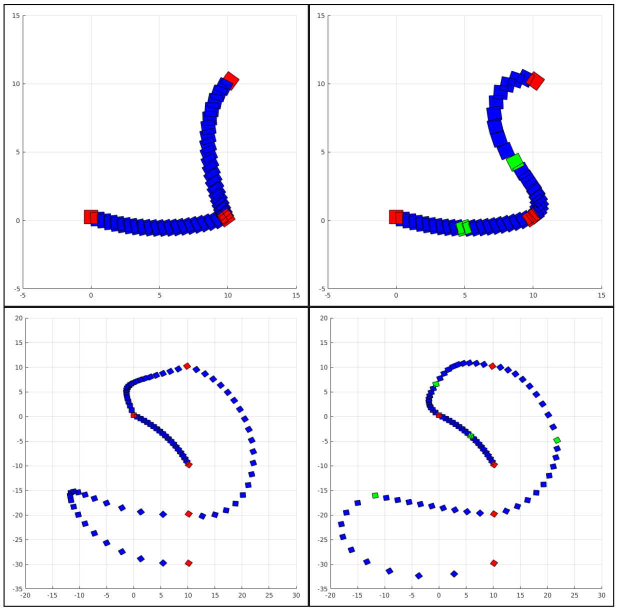

Fig. 8. Subdivision scheme for smooth interpolation of multiple affine transformations. A sequence of affine transformation is provided (shown in red). Left column: Each pair of consecutive affine transformations sequence are interpolated for a time step of $dt = .05$ (shown in blue). Right column: In addition to the initial sequence of transformations additional intermediate transformation from column 1 (shown in green) are provided for interpolation. Each pair in the new affine transformation sequences are interpolated for a time step of $dt = .1$ As can be seen, the interpolation becomes smoother after providing additional transformations.

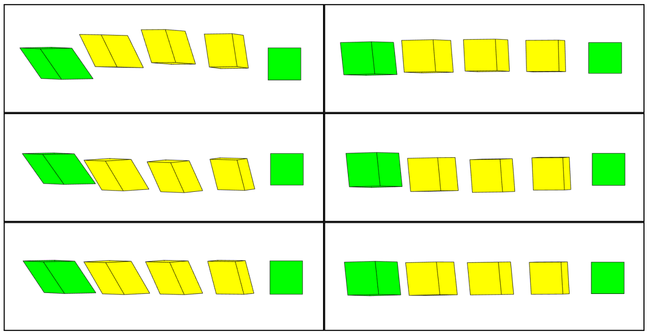

Fig. 10. Interpolations corresponding to three vertex orderings(row-wise) for two deformations obtained by varying vertices on the circle. The two columns demonstrate results for transformations containing decreasing degree of shear. Interpolated transformations are shown for $t = .25, .5, .75$. It is evident that the path change is higher for a higher shear component. More results are provided in the supplementary material.

A.1.2 (Theorem 1).

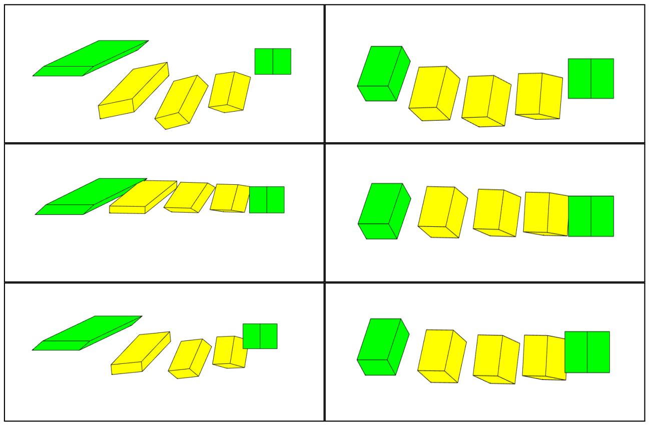

Fig. 11. Interpolations corresponding to three vertex orderings(row-wise) for two deformations obtained by translating vertices on the coordinate axes. The two columns demonstrate results for transformations containing decreasing degree of shear. Interpolated transformations are shown for $t = .25, .5, .75$. It is evident that the path change is higher for a higher shear component. More results are provided in the supplementary material.

(1) An affine transformation $\mathbb{T}$ can be decomposed into a translation and a linear component, $\mathbb{T} = [L \mid d]$. The translation part d aligns the two centroids and is thus uniquely determined. The linear part can then be written as a map of the difference vectors of vertices with the centroid from one tetrahedron to corresponding difference vectors in the other tetrahedron. Since the tetrahedrons are non-degenerate, these two sets of difference vectors are linearly independent, implying the uniqueness of L. Moreover, since the two tetrahedrons have the same orientation, $det(\mathbb{T}) > 0$.

(2) The translation part is invertible, while the linear part is invertible as far as the tetrahedrons are non-degenerate. Hence $\mathbb{T}^{-1}$ exists. Fixing tetrahedron Δ_Y, tetrahedron Δ_X is given by $\Delta_X = \mathbb{T}^{-1}\Delta_Y$ or fixing tetrahedron Δ_X, tetrahedron Δ_Y is given by $\Delta_Y = \mathbb{T}\Delta_X$.

(3) We will first prove the statement for the case where Δ_Y is assumed to be the canonical tetrahedron, and later generalize it for any arbitrary tetrahedron. In order to write the affine transformation $\mathbb{T}$ as a unique product of elements from $SE(3), G_S$ and G_A, we show that the three components are unique. To begin with, there exists a unique $SE(3)$ element E that does the rigid alignment of the tetrahedron to the tetrahedron with vertices $(\mathbf{0}, v_1, v_2, v_3)$ (refer to Figure 1), since there exists a unique rotation matrix that aligns two orthogonal vectors, and a unique translation vector that aligns two points in $\mathbb{R}^3$. Then, a unique uniform scaling (in $\mathbb{R}^3$) that aligns the second vertex, say $v_1 = (v_{1_x}, 0, 0)$ to the vector $(1,0,0)$ is given by $s = \dfrac{1}{v_{1_x}}$. Let the third and fourth vertex coordinates after the uniform scaling be $(w_{2_x}, w_{2_y}, 0)$ and $(w_{3_x}, w_{3_y}, w_{3_z})$, respectively. The shear component can be computed by solving the linear system of equations: $A[w_2 \;\; w_3] = [(1\;1\;0)^T \;\; (0\;0\;1)^T]$. The solution is uniquely given by,

$$\alpha_1 = \frac{1 - w_{2_x}}{w_{2_y}}, \; \alpha_2 = \frac{1}{w_{2_y}},$$

$$\alpha_3 = -\frac{(1 - w_{2_x})w_{3_y} + w_{3_x} w_{2_y}}{w_{2_y} w_{3_z}}, \tag{8}$$

$$\alpha_4 = -\frac{w_{3_y}}{w_{2_y} w_{3_z}}, \; \alpha_5 = \frac{1}{w_{3_z}}. \tag{9}$$

Note that since the tetrahedron has been assumed to be non-degenerate and has the same orientation as the canonical tetrahedron, $w_{2_y} > 0$ and $w_{3_z} > 0$. Thus, the orientation-preserving affine transformation $\mathbb{T}$ can be decomposed into components (E, A, S) uniquely, and $\mathbb{T} = ASE$.

For the case where Δ_Y is not the canonical tetrahedron, let Δ represent the canonical tetrahedron. As discussed earlier, there exist unique orientation-preserving affine transformations $\mathbb{T}_X$ and $\mathbb{T}_Y$ such that $\mathbb{T}_X\Delta_X = \mathbb{T}_Y\Delta_Y = \Delta$. Since $\mathbb{T}\Delta_X = \Delta_Y$ is unique, $\mathbb{T} = (\mathbb{T}_Y)^{-1}\mathbb{T}_X$, and thus $\mathbb{T}$ also maps the tetrahedron $(\mathbb{T}_X)^{-1}\mathbb{T}_Y\Delta$ to Δ, thus getting us back to the previous case.

A.1.3 (Lemma 1 - Steady Interpolation). In the case where the affine transformation consists of a single component, from Equation (2) and the identity $\exp(t \log M) = \exp(\log(M^t)) = M^t$, it is clear that $\mathbb{T}_t = \mathbb{T}^t$. When $\mathbb{T} = AS$ for some $A \in G_A$ and $s \in G_S$, since A and S commute,

$$\mathbb{T}_t = \exp(t \log A)\exp(t \log S) = A^t S^t = (AS)^t$$

Hence $\mathbb{T}_t = \mathbb{T}^t$.

A.1.4 (Lemma 2 - Volume Preservation). Let $\det(\mathbb{T}) = 1$. Since $\mathbb{T} = ASE$, we have $\det(ASE) = \det(A)\det(S)\det(E) = 1$. Using the fact that $\det(E) = \det(R) = 1$, $\det(S) = s^3$ and $\det(A) = \alpha_2\alpha_5$, we get $\alpha_2\alpha_5 s^3 = 1$. For any $t \in [0,1]$,

$$\begin{aligned}\det(\mathbb{T}_t) &= \det\left(\exp(t\log A)\exp(t\log S)\exp(t\log E)\right)\\ &= \det(\exp(t\log A))\det(\exp(t\log S))\det(\exp(t\log E))\\ &= \exp(t(\log\alpha_2 + \log\alpha_5))\exp(t\log s^3)\\ &= (\alpha_2\alpha_5 s^3)^t = 1,\end{aligned}$$

where the penultimate equation is obtained using the relation $\det(\exp(Q)) = \exp(tr(Q))$, with $tr(\cdot)$ denoting the trace operator, the form of $\log A$ given in Appendix A, and the fact that $tr(t \log E) = 0$.

A.1.5 (Lemma 3 - Monotonic variation of volume). In order to show monotonicity of the change of volume effected by the interpolated transformations, it is enough to show that the sign of $\frac{d}{dt}\det(\mathbb{T}_t), 0 < t < 1$, remains constant. Using the relation $\det(\mathbb{T}_t) = (\alpha_2\alpha_5 s^3)^t$ (derived in Lemma 2), we get

$$\begin{aligned}\frac{d}{dt}\det(\mathbb{T}_t) &= \frac{d}{dt}(\alpha_2\alpha_5 s^3)^t, 0 < t < 1\\ &= (\alpha_2\alpha_5 s^3)^t \log(\alpha_2\alpha_5 s^3).\end{aligned}$$

Since α_2, α_5, s and t are positive real numbers and $\det(\mathbb{T}) = \alpha_2\alpha_5 s^3$, we get $sign(\frac{d}{dt}\det(\mathbb{T}_t)) = sign(\log(\alpha_2\alpha_5 s^3))$

$= sign(\log(\det(\mathbb{T})))$. Thus $\forall t \in (0,1)$, $\frac{d}{dt}\det(\mathbb{T}_t) > 0 (< 0)$ if $det(\mathbb{T}) > 1 (< 1)$.

A.1.6 (Lemma 4 - Isometric Transformations). Since $\mathbb{T}$ is an isometry, it is a rigid transformation, i.e. $\mathbb{T} = E \in SE(3)$. The rigid transformation consists of a rotation $R \in SO(3)$ and a translation vector $d \in \mathbb{R}^3$. The interpolated transformation $\mathbb{T}^t, t \in [0,1]$ computed using Equation (2) is also a rigid transformation denoted by $\mathbb{T}^t = E^t \in SE(3)$ with a corresponding rotation R^t and translation vector $d^t \in \mathbb{R}^3$. It is easy to see that $||\mathbb{T}^t(x-y)|| = ||R^t(x-y)|| = ||x-y||$, proving that $\mathbb{T}^t$ is also an isometry, as required.

A.1.7 (Lemma 5 - Reversibility). Let the Lie group representations of the tetrahedrons $\Delta_p, \Delta_q, \Delta_p^q(t)$ be $p, q, r_p^q(t)$, respectively. Then $r_p^q(t) = p\exp(t\log(p^{-1}q))$, and

$$\begin{aligned} r_q^p(1-t) &= q\exp\left((1-t)\log(q^{-1}p)\right) \\ &= q\exp\left(\log(q^{-1}p)\right)\exp\left(-t\log(q^{-1}p)\right) \\ &= p\exp\left(t\log(p^{-1}q)\right) = r_p^q(t). \end{aligned}$$

Thus, our algorithm is reversible.

A.1.8 (Theorem 2 - Invariance to Canonical tetrahedron). Let Δ_p, Δ_q be the tetrahedrons to be interpolated, and let $\Delta, \Delta_1, \Delta_2$ be the original and two arbitrary canonical tetrahedrons.

Let (E_1, A_1, s_1) represent tetrahedron Δ_1 with respect to Δ_2, and let (E_2, A_2, s_2) represent tetrahedron Δ_2 with respect to Δ_1. Let $\Delta_{c_i}, i = 1,2$ be the interpolated tetrahedrons at some time t obtained when using canonical tetrahedrons $\Delta_i, i = 1,2$, with Lie group representations $(E_{c_i}, A_{c_i}, s_{c_i}), i = 1,2$, respectively. We now prove that the two representations $(E_{c_i}, A_{c_i}, s_{c_i})$, $i = 1,2$ represent the same tetrahedron. In the proposed framework (refer to Equation (4)), the components $E_{c_i}, A_{c_i}, s_{c_i}$, $i = 1,2$ are given as:

$$\begin{aligned} E_{c_i} &= E_{p_i}\exp(t\log(E_{p_i}^{-1}E_{q_i})), \\ A_{c_i} &= A_{p_i}\exp(t\log(A_{p_i}^{-1}A_{q_i})), \\ s_{c_i} &= s_{p_i}\exp(t\log(s_{p_i}^{-1}s_{q_i})). \end{aligned} \tag{10}$$

Since the rigid transformation component does not depend on the choice of canonical tetrahedron, $E_{p_1} = E_{p_2} = E_p$ and $E_{q_1} = E_{q_2} = E_q$. Then, one can conclude that $E_{c_1} = E_{c_2}$. Let l_p, l_q, l_1, l_2 denote the lengths of the first edge vectors of tetrahedrons $\Delta_p, \Delta_q, \Delta_1$ and Δ_2 respectively. Then, $s_{p_1}^{-1}s_{q_1} = \frac{l_p}{l_1}\frac{l_1}{l_q} = \frac{l_p}{l_2}\frac{l_2}{l_q} = s_{p_2}^{-1}s_{q_2}$.

Similarly by working out the shear matrices $A_{p_i}, i = 1,2$ and $A_{q_i}, i = 1,2$ in general, it can be shown that $(A_{p_i})^{-1}A_{q_i}, i = 1,2$ both depend only on the coordinates of the tetrahedrons Δ_p and Δ_q, and hence are equal. The above facts show that the tangent vector at identity between Lie group representations of tetrahedrons Δ_p and Δ_q is invariant to the choice of canonical tetrahedron.

The transformations between Δ_p and Δ_1, Δ_2 are

$$\begin{aligned} (E_1^{-1}A_{p_1}S_{p_1}E_p)\Delta_p &= \Delta_1 \\ (E_2^{-1}A_{p_2}S_{p_2}E_p)\Delta_p &= \Delta_2, \end{aligned}$$

respectively. Using

$$(E_2^{-1}A_1S_1E_1)\Delta_1 = \Delta_2, \tag{11}$$

in the equation above gives us

$$A_1S_1A_{p_1}S_{p_1} = A_{p_2}S_{p_2}.$$

Similarly, the transformations between the two interpolated tetrahedrons and their respective canonical tetrahedrons are

$$(E_1^{-1}A_{c_1}S_{c_1}E_{c_1})^{-1}\Delta_1 = \Delta_{c_1} \tag{12}$$

$$(E_2^{-1}A_{c_2}S_{c_2}E_{c_2})^{-1}\Delta_2 = \Delta_{c_2}. \tag{13}$$

Substituting Equation (11) in Equation (13) above yields

$$(E_1^{-1}S_1^{-1}A_1^{-1}A_{c_2}S_{c_2}E_{c_2})^{-1}\Delta_1 = \Delta_{c_2}. \tag{14}$$

Since $E_{c_1} = E_{c_2}$, in order to prove $\Delta_{c_1} = \Delta_{c_2}$, it remains to show that

$$A_1S_1A_{c_1}S_{c_1} = A_{c_2}S_{c_2}. \tag{15}$$

Using the definitions of the transformations from Equations (10) on the left hand side, and the fact that the diagonal scaling matrices S commute with the shear matrices A (both embedded in 4×4 matrices), we get the desired result:

$$\begin{aligned} &A_1S_1A_{c_1}S_{c_1} \\ &= A_1S_1A_{p_1}\exp(t\log(A_{p_1}^{-1}A_{q_1}))\,S_{p_1}\exp(t\log(S_{p_1}^{-1}S_{q_1})) \\ &= A_1S_1A_{p_1}S_{p_1}\exp(t\log(A_{p_1}^{-1}A_{q_1}))\exp(t\log(S_{p_1}^{-1}S_{q_1})) \\ &= A_{p_2}S_{p_2}\exp(t\log(A_{p_1}^{-1}A_{q_1}))\exp(t\log(S_{p_1}^{-1}S_{q_1})) \\ &= A_{p_2}\exp(t\log(A_{p_2}^{-1}A_{q_2}))\,S_{p_2}\exp(t\log(S_{p_2}^{-1}S_{q_2})) \\ &= A_{c_2}S_{c_2} \end{aligned}$$

Hence both representations $(E_{c_1}, A_{c_1}, s_{c_1})$ and $(E_{c_2}, A_{c_2}, s_{c_2})$ are representation of the same tetrahedron and the interpolation process remains invariant to the role of canonical tetrahedron.

A.2 Matrix Exponential and Logarithm maps for $SE(3), G_A$ and G_S

A.2.1 Exponential map on $SE(3)$. Let $A = \begin{bmatrix} \Omega & t \\ 0_3^T & 0 \end{bmatrix} \in \mathfrak{se}(3)$. If $||\Omega||_F = 0$ then $\exp(A) = \begin{bmatrix} I_3 & t \\ 0_3^T & 1 \end{bmatrix}$, else $\exp(A) = \begin{bmatrix} \exp(\Omega) & Vt \\ 0_3^T & 1 \end{bmatrix}$, where $\exp(\Omega) = I_3 + \frac{\sin\theta}{\theta}\Omega + \frac{(1-\cos\theta)}{\theta^2}\Omega^2$ and $V = I_3 + \frac{(1-\cos\theta)}{\theta^2}\Omega + \frac{(\theta-\sin\theta)}{\theta^3}\Omega^3$, with $\theta = \sqrt{-\frac{1}{2}Tr(\Omega^2)}$.

A.2.2 Logarithm map on $SE(3)$. Let $A = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \in SE(3)$. $\log(A) = \begin{bmatrix} \log(R) & V^{-1}t \\ 0_3^T & 0 \end{bmatrix}$, where $\log(R) = \dfrac{\theta}{2sin\theta}\left(R - R^T\right)$ and $V = I_3 + \frac{(1-\cos\theta)}{\theta^2}R + \frac{(\theta-\sin\theta)}{\theta^3}R^3$, with $\theta = acos\left(\dfrac{Tr(R)-1}{2}\right)$.

A.2.3 Exponential map on G_A. Let $A = \begin{bmatrix} 0 & \bar{\alpha}_1 & \bar{\alpha}_3 \\ 0 & \bar{\alpha}_2 & \bar{\alpha}_4 \\ 0 & 0 & \bar{\alpha}_5 \end{bmatrix} \in \mathfrak{g}_A$.

$$
Exp(A) = \begin{cases}
\begin{bmatrix} 1 & \bar{\alpha}_1 & \dfrac{\bar{\alpha}_1\bar{\alpha}_4\exp(\bar{\alpha}_5) + \bar{\alpha}_3\bar{\alpha}_5\exp(\bar{\alpha}_5) - \bar{\alpha}_1\bar{\alpha}_4\bar{\alpha}_5 - \bar{\alpha}_1\bar{\alpha}_4 - \bar{\alpha}_3\bar{\alpha}_5}{\bar{\alpha}_5{}^2} \\ 0 & 1 & \dfrac{\bar{\alpha}_4(\exp(\bar{\alpha}_5) - 1)}{\bar{\alpha}_5} \\ 0 & 0 & \exp(\bar{\alpha}_5) \end{bmatrix}, & \text{if } \bar{\alpha}_2 = 0 \text{ and } \bar{\alpha}_5 \neq 0 \\
\begin{bmatrix} 1 & \dfrac{\bar{\alpha}_1(exp(\bar{\alpha}_2) - 1)}{\bar{\alpha}_2} & \dfrac{\bar{\alpha}_1\bar{\alpha}_4\exp(\bar{\alpha}_2) - \bar{\alpha}_1(1 + \bar{\alpha}_2)\bar{\alpha}_4 + \bar{\alpha}_2{}^2\bar{\alpha}_3}{\bar{\alpha}_2{}^2} \\ 0 & \exp(\bar{\alpha}_2) & \dfrac{\bar{\alpha}_4(\exp(\bar{\alpha}_2) - 1)}{\bar{\alpha}_2} \\ 0 & 0 & 1 \end{bmatrix}, & \text{if } \bar{\alpha}_5 = 0 \text{ and } \bar{\alpha}_2 \neq 0 \\
\begin{bmatrix} 1 & \bar{\alpha}_1 & \bar{\alpha}_3 + \dfrac{\bar{\alpha}_1\bar{\alpha}_4}{2} \\ 0 & 1 & \bar{\alpha}_4 \\ 0 & 0 & 1 \end{bmatrix}, & \text{if } \bar{\alpha}_2 = 0 \text{ and } \bar{\alpha}_5 = 0 \\
\begin{bmatrix} 1 & \dfrac{\bar{\alpha}_1(\exp(\bar{\alpha}_2) - 1)}{\bar{\alpha}_2} & \dfrac{((\bar{\alpha}_1\bar{\alpha}_4 + \bar{\alpha}_3)\bar{\alpha}_2 - \bar{\alpha}_1\bar{\alpha}_4)\exp(\bar{\alpha}_2) + \bar{\alpha}_1\bar{\alpha}_4 - \bar{\alpha}_2\bar{\alpha}_3}{\bar{\alpha}_2{}^2} \\ 0 & \exp(\bar{\alpha}_2) & \bar{\alpha}_4\exp(\bar{\alpha}_2) \\ 0 & 0 & \exp(\bar{\alpha}_2) \end{bmatrix}, & \text{if } \bar{\alpha}_2 = \bar{\alpha}_5 \neq 0 \\
\begin{bmatrix} 1 & \dfrac{\bar{\alpha}_1(\exp(\bar{\alpha}_2) - 1)}{\bar{\alpha}_2} & \dfrac{(-\bar{\alpha}_2(\bar{\alpha}_1\bar{\alpha}_4 - \bar{\alpha}_2\bar{\alpha}_3 + \bar{\alpha}_3\bar{\alpha}_5)exp(\bar{\alpha}_5) + \bar{\alpha}_1\bar{\alpha}_4\bar{\alpha}_5\exp(\bar{\alpha}_2) + (\bar{\alpha}_2 - \bar{\alpha}_5)(\bar{\alpha}_1\bar{\alpha}_4 - \bar{\alpha}_2\bar{\alpha}_3))}{(\bar{\alpha}_2\bar{\alpha}_5(\bar{\alpha}_2 - \bar{\alpha}_5))} \\ 0 & \exp(\bar{\alpha}_2) & \bar{\alpha}_4\dfrac{(\exp(\bar{\alpha}_2) - \exp(\bar{\alpha}_5))}{(\bar{\alpha}_2 - \bar{\alpha}_5)} \\ 0 & 0 & \exp(\bar{\alpha}_5) \end{bmatrix}, & \text{otherwise.}
\end{cases}
$$

A.2.4 Logarithm map on G_A. Let $A = \begin{bmatrix}\begin{bmatrix}1 & \alpha_1 & \alpha_3 \\ 0 & \alpha_2 & \alpha_4 \\ 0 & 0 & \alpha_5\end{bmatrix}\end{bmatrix} \in G_A$.

$$log(A) = \begin{cases} \begin{bmatrix} 0 & \alpha_1 & \dfrac{\log(\alpha_5)(\alpha_3\alpha_5 - \alpha_3 + \alpha_1\alpha_4)}{(\alpha_5 - 1)^2} - \dfrac{\alpha_1\alpha_4}{(\alpha_5 - 1)} \\ 0 & 0 & \dfrac{\alpha_4 \log(\alpha_5)}{(\alpha_5 - 1)} \\ 0 & 0 & \log(\alpha_5) \end{bmatrix}, \text{ if } \alpha_2 = 1 \text{ and } \alpha_5 \neq 1 \\ \begin{bmatrix} 0 & \dfrac{\alpha_1 \log(\alpha_2)}{\alpha_2 - 1} & \dfrac{\alpha_1\alpha_4 \log(\alpha_2)}{(\alpha_2 - 1)^2} - \dfrac{(\alpha_3 - \alpha_2\alpha_3 + \alpha_1\alpha_4)}{(\alpha_2 - 1)} \\ 0 & \log(\alpha_2) & \dfrac{\alpha_4 \log(\alpha_2)}{(\alpha_2 - 1)} \\ 0 & 0 & 0 \end{bmatrix}, \text{ if } \alpha_5 = 1 \text{ and } \alpha_2 \neq 1 \\ \begin{bmatrix} 0 & \alpha_1 & \alpha_3 - \dfrac{(\alpha_1\alpha_4)}{2} \\ 0 & 0 & \alpha_4 \\ 0 & 0 & 0 \end{bmatrix}, \text{ if } \alpha_2 = \alpha_5 = 1 \\ \begin{bmatrix} 0 & \dfrac{\alpha_1 \log(\alpha_2)}{\alpha_2 - 1} & \dfrac{\alpha_3 \log(\alpha_2)}{(\alpha_2 - 1)} + \dfrac{\alpha_1\alpha_4}{\alpha_2(\alpha_2 - 1)} - \dfrac{\alpha_1\alpha_4 \log(\alpha_2)}{(\alpha_2 - 1)^2} \\ 0 & \log(\alpha_2) & \dfrac{\alpha_4}{\alpha_2} \\ 0 & 0 & \log(\alpha_2) \end{bmatrix}, \text{ if } \alpha_2 = \alpha_5 \neq 1 \\ \begin{bmatrix} 0 & \dfrac{\alpha_1 \log(\alpha_2)}{\alpha_2 - 1} & \dfrac{\alpha_1\alpha_4 \log(\alpha_2)}{(\alpha_2 - \alpha_5)(\alpha_2 - 1)} - \dfrac{(\alpha_1\alpha_4 - \alpha_2\alpha_3 + \alpha_3\alpha_5)\log(\alpha_5)}{(\alpha_2 - \alpha_5)(\alpha_5 - 1)} \\ 0 & \log(\alpha_2) & \dfrac{\alpha_4 \log(\alpha_2)}{(\alpha_2 - \alpha_5)} - \dfrac{\alpha_4 \log(\alpha_5)}{(\alpha_2 - \alpha_5)} \\ 0 & 0 & \log(\alpha_5) \end{bmatrix}, \text{ otherwise} \end{cases}$$

A.2.5 Exponential and Logarithm map on G_S. Note that $G_S = \mathbb{R}^+$ and $\mathfrak{g}_S = \mathbb{R}$, therefore exponential and logarithm maps coincide with usual scalar exponential and logarithm maps.

A.2.6 Note. Exponential and Logarithm maps for A and s are converted into 4×4 matrices by embedding them in the top-left 3×3 block of the 4×4 identity matrix, whenever required to be composed with other transformations.