

Camera Pose Filtering with Local Regression Geodesics on the Riemannian Manifold of Dual Quaternions

Benjamin Busam^{1,2} Tolga Birdal^{1,3}
Nassir Navab^{1,4}

¹ Computer Aided Medical Procedures, Technische Universität München, Germany

² FRAMOS GmbH, Germany ³ Siemens AG, Germany

⁴ Computer Aided Medical Procedures, Johns Hopkins University, US

b.busam@framos.com, tolga.birdal@tum.de, navab@cs.tum.edu

Abstract

Time-varying, smooth trajectory estimation is of great interest to the vision community for accurate and well behaving 3D systems. In this paper, we propose a novel principal component local regression filter acting directly on the Riemannian manifold of unit dual quaternions \mathbb{DH}_1 . We use a numerically stable Lie algebra of the dual quaternions together with exp and log operators to locally linearize the 6D pose space. Unlike state of the art path smoothing methods which either operate on $SO(3)$ of rotation matrices or the hypersphere \mathbb{H}_1 of quaternions, we treat the orientation and translation jointly on the dual quaternion quadric in the 7-dimensional real projective space \mathbb{RP}^7 . We provide an outlier-robust IRLS algorithm for generic pose filtering exploiting this manifold structure. Besides our theoretical analysis, our experiments on synthetic and real data show the practical advantages of the manifold aware filtering on pose tracking and smoothing.

1. Introduction

Many 3D computer vision tasks require a robust and reliable understanding of position and orientation in space and a outlier-free localization of the camera with respect to its surroundings is a fundamental requirement in machine vision tasks such as registration, reconstruction or tele-robotics. Independent on the given data modality, the natural underlying structure of the input is a temporally ordered set which can be analyzed sequentially. In an example scenario such as SLAM, egocentric vision or marker tracking, a single camera provides a stream of consecutive images generating a pose path. In this paper, we address motion smoothing, where - given the per frame motion es-

timates - the goal is to synthesize a new camera trajectory, which is smooth and closer to the underlying movement or intended trajectory.

1.1. Pose Parametrization

Different parametrizations are prominent to accurately and efficiently describe spatial displacements. Most widely used representations include homogeneous matrices, quaternions or twist-coordinates. While many of these are sufficient to handle orientation, the translational component is usually treated separately [10, 15]. Moreover, the algorithms which use the semi-direct product of $SO(3)$ and \mathbb{R}^3 to describe elements of the group $SE(3)$ suffer from the structural embedding of the rotational part in an higher dimensional space. One solution to circumvent this problem emerges with the dual quaternion (DQ) parametrization, in which all pose components form a common space. Besides a smaller memory footprint the representation has the advantage of being cheaper in case of consecutive transformations while it is numerically more stable to calculate due to the fact that the matrix space is higher dimensional and a re-orthogonalization [3] is computationally much more expensive than a re-normalization. The DQ formulation has already been successful in applications of interpolation [5], skinning [17] or rigid body dynamics [37]. Unfortunately, the applications are still not numerous. This is partially due to the fact that the manifold is more complex than the one of the quaternions and immediate geometric intuitions are lacking. With this paper, we devise a general picture on how to operate on the quadric shaped Riemannian manifolds of dual quaternions by studying the exponential and logarithm maps and use them to construct an outlier aware, robust pose smoothing algorithm.

1.2. Manifold Filter

In practice, acquired camera poses follow complex noise models as partially interdependent factors such as sensor noise, hand-shake, wrong pose estimates and velocity appear. The behaviour is highly non-linear and tedious to model directly. Thus, the requirement of tuning the model parameters for Kalman filter methods is not suited for certain applications. We employ a non-parametric smoother without explicit distribution assumptions on the overall data which we embed in a framework on the DQ-manifold. At this point, the local regression (LR) reveals itself to be the method of choice. In the simplest form of LR, one of the points is treated as the center and a linear relationship between the independent and response variables in its local neighbourhood are sought. This point is then projected onto the line, gaining a new, smoothed position. This procedure iterates over the sequence, while iteratively denoising the points. While this procedure is fairly easy for Euclidean spaces, many of these operations do not generalize to non-Euclidean manifolds. We make use of the fact that a point on the pose path is selected and the regression considers only the immediate neighbourhood. Hence, we utilize the exponential and logarithm maps in the quaternion notation to operate on the local tangent space, alleviating the tedious non-Euclidean geometry. We contribute a PCA based revised LR formulation on the local linearized space, making effective use of the Riemannian DQ-manifold.

Our experiments show that, while a weighted manifold-PC regression is mostly satisfactory, for pose sequences with outliers, IRLS smoothing on DQ outperforms in terms of fidelity. The advantage of manifold smoothing also reveals as the noise level increases.

2. Related Literature

The Clifford Algebra of dual quaternions [8] extends the non-commutative division algebra of the quaternions \mathbb{H} [14]. Besides pure [2] and applied geometry [29], quaternions have a broad history in computer vision [28].

Rotations are more efficiently concatenated if a quaternion representation is chosen and effects such as gimbal lock are avoided [21], which is why quaternions are applied to various real-time 3D processing pipelines [23]. In particular the fact that \mathbb{H} can be identified with points on a 3-dimensional hypersphere S^3 makes it attractive for animation and rendering as SLERP [31] interpolation can be calculated efficiently.

In spite of their usefulness for many real-time calculations, dual approaches did not receive the same attention. A dual quaternion $\mathbf{Q} \in \mathbb{DH}_1$ of unit length can be used to represent a rigid body displacement of an object. Kuang [20] uses this to show the advantages of dual quaternions for real-time motion animation for clothed body

movements and Kavan [17] applied \mathbb{DH}_1 for skinning. Dual methods are also present in blending [27] and complex hierarchical rigid body transforms [18] and more recently, movement extrapolation [5] has been an application field for \mathbb{DH}_1 .

We represent rigid body movements as trajectories on \mathbb{DH}_1 where we filter noisy paths. Regression models for interpolation are essential for computer animation and refinement of pose data [26]. Kavan [16] approximates the geodesic distance with an L2-norm in \mathbb{R}^8 for real-time transformation blending. However, the unit dual quaternion space is non-Euclidean.

A diffusion approach in this regard has been proposed by Torsello [33] where the Riemannian metric is minimized for multiview registration. We also take the Riemannian nature of \mathbb{DH}_1 into account and perform a differential correction by moving to the tangent space where we apply a local smoothing method. Similar filtering approaches have been studied extensively [10, 15] in the joint space $\mathbb{H}_1 \times \mathbb{R}^3$, where the pose is split into rotation and translation parts. Ng [25] proposes a Gaussian smoothing on the non-dual quaternion manifold [25] while Srivatsan [32] explicitly assumes an underlying noise model and uses a Linear Kalman filter on dual quaternions in $SE(3)$ which is done with an Extended Kalman Filter (EKF) for pose estimation by Filipe [12]. On the application side, smoothing techniques are used for Video Stabilization [15] and robotics [10].

3. Mathematical Formulation

A substantial part of the proposed framework uses approaches from differential geometry which are applied to the space of poses. We first introduce the pose spaces \mathbb{H}_1 and \mathbb{DH}_1 of unit (dual) quaternions which we analyze thereafter locally and conclude by giving the exponential and logarithm maps in quaternion representation such that these can be implemented in the proposed algorithms afterwards. The notation follows the convention of Busam [5] where the algebraic structures are presented in more detail.

3.1. Quaternions

Quaternions extend the complex numbers with three imaginary units $\mathbf{i}, \mathbf{j}, \mathbf{k}$.

Definition 1

A *quaternion* q is an element of the algebra \mathbb{H} in the form

$$q = q_1\mathbf{1} + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} = (q_1, q_2, q_3, q_4)^T, \quad (1)$$

with $(q_1, q_2, q_3, q_4)^T \in \mathbb{R}^4$ and $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -\mathbf{1}$.

We also write $\mathbf{q} := [a, \mathbf{v}]$ with the scalar part $a = q_1 \in \mathbb{R}$ and the vector part $\mathbf{v} = (q_2, q_3, q_4)^T \in \mathbb{R}^3$. The conjugate $\bar{\mathbf{q}}$ of the quaternion \mathbf{q} is given by

$$\bar{\mathbf{q}} := q_1 - q_2\mathbf{i} - q_3\mathbf{j} - q_4\mathbf{k}. \quad (2)$$

Quaternions are of particular interest in computer vision due to their connection with spatial rotations [11]. A **unit quaternion** $\mathbf{q} \in \mathbb{H}_1$ with

$$1 \stackrel{!}{=} \|\mathbf{q}\| := \mathbf{q} \cdot \bar{\mathbf{q}} \quad (3)$$

gives a compact and numerically stable parametrization to represent orientation and rotation of objects in \mathbb{R}^3 which avoids gimbal lock [21].

Rotations with Quaternions

The rotation around the unit axis $\mathbf{v} = (v_1, v_2, v_3)^T \in \mathbb{R}^3$ with angle θ is thereby given by

$$\mathbf{r} = [\cos(\theta/2), \sin(\theta/2)\mathbf{v}]. \quad (4)$$

Identifying antipodal points \mathbf{q} and $-\mathbf{q}$ with the same element in $SO(3)$, the unit quaternions form a double covering group of the 3D rotations and any **pure quaternion**

$$\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}. \quad (5)$$

of the point $\mathbf{u} = (x, y, z)^T \in \mathbb{R}^3$ is rotated by the unit quaternion \mathbf{r} via the sandwiching product map

$$\mathbf{p} \mapsto \mathbf{r} \cdot \mathbf{p} \cdot \bar{\mathbf{r}}. \quad (6)$$

3.2. Dual Quaternions

Similar to the representation of rotations by quaternions, we can use **dual quaternions** of unit length to represent spatial displacements. We can define a dual quaternion as an ordered pair of quaternions with dual numbers as coefficients. A **dual number** Z is an element of the algebra \mathbb{D} that can be written [19] as $Z = r + \varepsilon s$ where $r, s \in \mathbb{R}$ and $\varepsilon^2 = 0$, where r is the real-part, s is the dual part, and ε is called the dual operator. The dual conjugate is similar to the complex conjugate of $\mathbb{R} + i\mathbb{R}$. It is given by $\hat{Z} := r - \varepsilon s$.

Extending this concept to quaternions, we can define dual quaternions.

Definition 2

A **dual quaternion** $\mathbf{Q} \in \mathbb{DH}$ is an ordered set of quaternions

$$\mathbf{Q} = \mathbf{r} + \varepsilon \mathbf{s} = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8)^T, \quad (7)$$

where $\mathbf{r}, \mathbf{s} \in \mathbb{H}$, $(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8)^T \in \mathbb{R}^8$ and

$$\varepsilon^2 = 0, \quad \varepsilon \mathbf{i} = \mathbf{i} \varepsilon, \quad \varepsilon \mathbf{j} = \mathbf{j} \varepsilon, \quad \varepsilon \mathbf{k} = \mathbf{k} \varepsilon. \quad (8)$$

The Clifford algebra of dual quaternions contains the real numbers \mathbb{R} , the complex numbers \mathbb{C} , the dual numbers \mathbb{D} , and the quaternions \mathbb{H} as sub-algebras. With the conjugate $\bar{\mathbf{Q}}$ of the dual quaternion $\mathbf{Q} = \mathbf{r} + \varepsilon \mathbf{s}$

$$\bar{\mathbf{Q}} := \bar{\mathbf{r}} + \varepsilon \bar{\mathbf{s}}, \quad (9)$$

we can study the constraints given for unit **unit dual quaternions** $\mathbf{Q} \in \mathbb{DH}_1$. If we demand unit length, it holds

$$1 \stackrel{!}{=} \|\mathbf{Q}\| := \mathbf{Q} \cdot \bar{\mathbf{Q}} = \mathbf{r}\bar{\mathbf{r}} + \varepsilon(\mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}}), \quad (10)$$

which gives the two distinct constraints

$$\mathbf{r}\bar{\mathbf{r}} = 1 \quad \text{and} \quad \mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}} = 0. \quad (11)$$

Displacements with Dual Quaternions

The unit dual quaternions are isomorphic to the group of rigid body displacement $SE(3)$ [1] and the two constraints (11) reduce the eight parameters of the dual quaternions to the six degrees of freedom of a rigid motion in space with its translation and rotation. If we write the translation as a pure quaternion \mathbf{t} (5) and the rotation as a unit quaternion \mathbf{r} (4), we can construct the unit dual quaternion

$$\mathbb{DH}_1 \ni \mathbf{Q} = \mathbf{r} + \varepsilon \frac{1}{2} \mathbf{t} \mathbf{r}. \quad (12)$$

Analogously to the quaternions, we formulate the **dual pure quaternion** \mathbf{P} for the point $\mathbf{p} = [0, \mathbf{u}]$ as $\mathbf{P} = \mathbf{1} + \varepsilon \mathbf{u}$ and the spatial displacement becomes the sandwiching product map on dual quaternions

$$\mathbf{P} \mapsto \mathbf{Q} \cdot \mathbf{P} \cdot \hat{\mathbf{Q}} = 1 + \varepsilon(\mathbf{r}\bar{\mathbf{u}} + \mathbf{t}), \quad (13)$$

where the conjugates for the dual quaternion and the dual are calculated consecutively.

3.3. Riemannian Geometry

The spaces \mathbb{H}_1 and \mathbb{DH}_1 can also be considered as differentiable Riemannian manifolds \mathbb{G} [34]. A continuous collection of inner products on the tangent space of \mathbb{G} at $\mathbf{x} \in \mathbb{G}$ defines a Riemannian metric. The shortest path on the manifold defined by such a metric is called the geodesic.

With these concepts, we take a closer look to the geometry of the (dual) quaternion space and calculate a specific mapping into the tangent space which we generalize with the help of parallel transport. This will pave the way to subsequently define different local geodesic regressors in pose space.

3.3.1 Geometry of \mathbb{H}_1 and \mathbb{DH}_1

With constraint (3), the unit quaternions form the three dimensional hypersphere $S^3 \in \mathbb{R}^4$. Thus \mathbb{H}_1 is isomorphic to the real projective space \mathbb{RP}^3 . Looking at the two constraints from (11), we can analyze the structure of the unit dual quaternion space. The first equation $\|\mathbf{r}\| = 1$ forces the real part \mathbf{r} of \mathbf{Q} to be of unit length, hence $\mathbf{r} \in \mathbb{H}_1$. This gives the 7-dimensional hypersphere $S^7 \in \mathbb{R}^8$ and the identification of antipodal points forms the seven dimensional

real projective space \mathbb{RP}^7 . The second equation reads as $\mathbf{r}\bar{\mathbf{s}} = -\mathbf{s}\bar{\mathbf{r}}$ and thus defines a quadric in \mathbb{RP}^7 . Thus \mathbb{DH}_1 is not a hypersphere. This need to be considered for any operation on the manifold.

3.3.2 Lie Groups and Parallel Transport

From a differential geometry perspective, a Lie group can be viewed as a differentiable Riemannian manifold. The Lie algebra to the Lie group is the tangent space at the identity of the group. Thus it gives a linearization of the Lie group near the identity. The map from the tangent space $T_{\mathbf{x}}\mathbb{G}$ at \mathbf{x} to the Lie group \mathbb{G} is called the exponential map

$$\exp_{\mathbf{x}} : T_{\mathbf{x}}\mathbb{G} \rightarrow \mathbb{G}, \quad (14)$$

which is locally defined and maps a vector in the tangent space to a point on the manifold following the geodesic on \mathbb{G} through \mathbf{x} . Its inverse is called the logarithm map

$$\log_{\mathbf{x}} : \mathbb{G} \rightarrow T_{\mathbf{x}}\mathbb{G}. \quad (15)$$

The mapping $\exp_{\mathbf{x}}$ at point $\mathbf{x} \in \mathbb{G}$ can be computed by parallel transport [13] as illustrated in Figure 1. With the exponential map $\exp_{\mathbf{1}} =: \exp$ at the identity $\mathbf{1} \in \mathbb{G}$ and the logarithm map it holds

$$\exp_{\mathbf{x}}(\mathbf{s}) = \mathbf{x} \exp(\mathbf{x}^{-1}\mathbf{s}), \quad (16)$$

$$\log_{\mathbf{x}}(\mathbf{q}) = \mathbf{x} \log(\mathbf{x}^{-1}\mathbf{q}). \quad (17)$$

As a next step, we want to use this to derive the exponential and logarithm maps at the identity for the elements of the groups $SO(3)$ and $SE(3)$ in quaternion notation. For matrices these maps are well studied objects [35], [24]. We study the exponential maps directly in (dual) quaternion space using its definition as a Maclaurin series.

3.3.3 Exponential and Logarithm map in \mathbb{H}

The identity in \mathbb{H}_1 is given by $\mathbf{1} = (1, 0, 0, 0)^T$. The tangent space $T_{\mathbf{1}}\mathbb{H}_1$ is thus the hyperplane to the hypersphere $S^3 \in \mathbb{R}^4$ and parallel to the axes x_2, x_3, x_4 passing through $\mathbf{1}$. Any quaternion in $T_{\mathbf{1}}\mathbb{H}_1$ is of the form

$$\mathbb{H} \ni \mathbf{q} = [0, \phi \mathbf{v}] \quad (18)$$

with $\mathbf{v} \in \mathbb{R}^3$, $\|\mathbf{v}\| = 1$ and the series writes as

$$\exp : T_{\mathbf{1}}\mathbb{H}_1 \rightarrow \mathbb{H}_1 \quad (19)$$

$$\mathbf{q} \mapsto 1 + \sum_{k=1}^{\infty} \frac{\mathbf{q}^k}{k!} := \sum_{k=0}^{\infty} \frac{\mathbf{q}^k}{k!} \quad (20)$$

$$= \cos(\phi) + \frac{\sin(\phi)}{\phi} \mathbf{q} \quad (21)$$

$$= [\cos(\phi), \sin(\phi) \mathbf{v}] =: \mathbf{r}. \quad (22)$$

where the second last step is done by recognizing the Taylor series for the sine and cosine function at 0. Note, that this relationship directly aligns with the notation in (4) while $\phi = \theta/2$ and the inverse function is given by

$$\log : \mathbb{H}_1 \rightarrow T_{\mathbf{1}}\mathbb{H}_1 \quad (23)$$

$$\mathbf{r} \mapsto [0, \phi \mathbf{v}]. \quad (24)$$

3.3.4 Exponential and Logarithm map in \mathbb{DH}

Let

$$\mathbb{DH} \ni \mathbf{Q} = \omega \mathbf{q} + \varepsilon \psi \mathbf{q}_{\varepsilon} \quad (25)$$

be a pure dual quaternion with the two pure quaternions $\mathbf{q}, \mathbf{q}_{\varepsilon} \in \mathbb{H}_1$. Simplification [30] of the Maclaurin series for the exponential map then yields

$$\exp : T_{\mathbf{1}}\mathbb{DH}_1 \rightarrow \mathbb{DH}_1 \quad (26)$$

$$\mathbf{Q} \mapsto \sum_{k=0}^{\infty} \frac{\mathbf{Q}^k}{k!} \quad (27)$$

$$= \frac{1}{2} (2 \cos(\omega) + \omega \sin(\omega)) \quad (28)$$

$$- \frac{1}{2\omega} (\omega \cos(\omega) - 3 \sin(\omega)) \mathbf{Q} \quad (29)$$

$$+ \frac{1}{2\omega} (\sin(\omega)) \mathbf{Q}^2 \quad (30)$$

$$- \frac{1}{2\omega^3} (\omega \cos(\omega) - \sin(\omega)) \mathbf{Q}^3. \quad (31)$$

Before we compute the inverse function, we make the observation that any unit dual quaternion

$$\mathbf{Q} = [\phi, \mathbf{v}] + \varepsilon [\phi_{\varepsilon}, \mathbf{v}_{\varepsilon}] =: [\Phi, \mathbf{V}] \quad (32)$$

with the dual entities

$$\Phi = \phi + \phi_{\varepsilon} \varepsilon \quad (33)$$

$$\mathbf{V} = \mathbf{v} + \mathbf{v}_{\varepsilon} \varepsilon. \quad (34)$$

can be written [9] equivalently to (4). For this, we calculate the dual trigonometric operators through a series expansion which brings

$$\sin(\Phi) := \sin(\phi) + \varepsilon \phi_{\varepsilon} \cos(\phi) \quad (35)$$

$$\cos(\Phi) := \cos(\phi) - \varepsilon \phi_{\varepsilon} \sin(\phi). \quad (36)$$

We proof the following lemma by explicit calculation of the dual quaternion representation.

Lemma 1

Any unit dual quaternion $\mathbf{Q} \in \mathbb{DH}_1$ can be written as

$$\mathbf{Q} = [\cos(\Theta/2), \sin(\Theta/2) \mathbf{V}], \quad (37)$$

where $\mathbf{V} \in \mathbb{DH}$ is a pure dual quaternion of form (25).

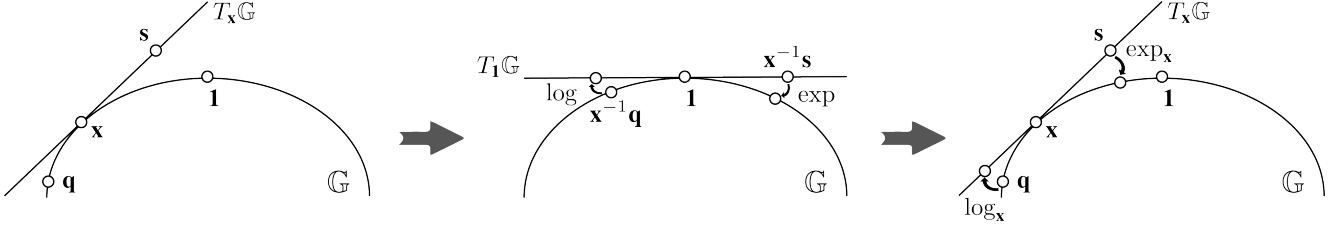


Figure 1. Parallel transport for the calculation of \exp_x and \log_x . The exponential and logarithm maps at the support point x are indirectly calculated via the explicit maps at the identity.

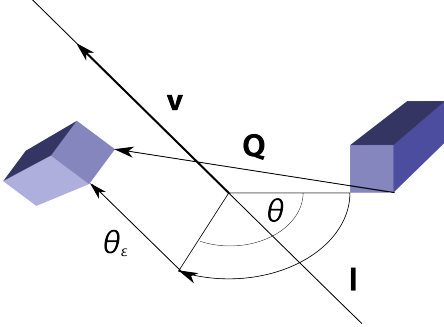


Figure 2. Screw linear displacement of rigid body with dual quaternion \mathbf{Q} along screw axis \mathbf{l} with angle θ and pitch θ_ε in the direction of \mathbf{v} .

Proof. Analogously to the quaternion rotation, the formulation (37) can be understood as a parametrization of the rigid body motion. According to Chasles' Theorem [7], a displacement can be modeled via a translation along a unique axis with a simultaneous rotation about the same axis. This is visualized in Figure 2. We construct the dual quaternion displacement for this motion explicitly in the form (37).

Let a rigid body transformation be given by a translation $\mathbf{t} \in \mathbb{R}^3$ and a rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ around the axis \mathbf{v} with $\|\mathbf{v}\| = 1$ with angle θ . From (12) we know already the unit dual quaternion for this displacement.

The parameters for the screw motion are angle θ , pitch θ_ε , screw axis \mathbf{l} with moment \mathbf{v}_ε (i.e. $\mathbf{v}_\varepsilon = \mathbf{p} \times \mathbf{v} \forall \mathbf{p} \in \mathbf{l}$) and direction \mathbf{v} . The angle θ is directly given. We first compute the pitch θ_ε in the direction \mathbf{v} of the axis as the projection of the translation onto the axis. This is $\theta_\varepsilon = \mathbf{t}^T \mathbf{v}$. In order to recover the moment \mathbf{v}_ε , we pick a point \mathbf{u} on the axis. With this we can describe \mathbf{t} in terms of θ_ε , \mathbf{v} , \mathbf{R} and \mathbf{u} as

$$\mathbf{t} = \theta_\varepsilon \mathbf{v} + (\mathbf{I} - \mathbf{R}) \mathbf{u} \quad (38)$$

and with the Rodrigues formula it holds

$$\mathbf{R}\mathbf{u} = \mathbf{u} + \sin(\theta) \mathbf{v} \times \mathbf{u} + (1 - \cos(\theta)) \mathbf{v} \times (\mathbf{v} \times \mathbf{u}). \quad (39)$$

Thus substituting this into (38) gives with $\mathbf{u}^T \mathbf{v} = 0$

$$\mathbf{u} = \frac{1}{2} \left(\mathbf{t} - (\mathbf{t}^T \mathbf{v}) \mathbf{v} + \cot\left(\frac{\theta}{2}\right) \mathbf{v} \times \mathbf{t} \right), \quad (40)$$

which brings for the moment vector

$$\mathbf{v}_\varepsilon = \mathbf{u} \times \mathbf{v} = \frac{1}{2} \left(\mathbf{t} \times \mathbf{v} + \cot\left(\frac{\theta}{2}\right) \mathbf{v} \times (\mathbf{t} \times \mathbf{v}) \right). \quad (41)$$

Substituting the rotation quaternion $\mathbf{r} = [q_0, \mathbf{q}]$ and using $\theta_\varepsilon = \mathbf{t}^T \mathbf{v}$ yields

$$\sin\left(\frac{\theta}{2}\right) \mathbf{v}_\varepsilon + \frac{\theta_\varepsilon}{2} \cos\left(\frac{\theta}{2}\right) \mathbf{v} = \frac{1}{2} (\mathbf{t} \times \mathbf{q} + q_0 \mathbf{t}), \quad (42)$$

which is the pure quaternion of the dual part in (12). Thus

$$\mathbf{Q} = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \mathbf{v}_\varepsilon \right] \quad (43)$$

$$+ \varepsilon \left[-\frac{\theta_\varepsilon}{2} \sin\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \mathbf{v}_\varepsilon + \frac{\theta_\varepsilon}{2} \cos\left(\frac{\theta}{2}\right) \mathbf{v} \right] \quad (44)$$

which equals (37) if we apply the trigonometric operators (35), (36) and the dual entity representations (33), (34). \square

We note that this representation separates the line information of the screw axis from the pitch and angle values in an algebraical way where the dual vector \mathbf{V} represents the axis of a screw motion with its direction vector and the dual angle Θ contains both the translation length and the angle of rotation.

Since for the exponential of a dual quaternion of the form $\mathbf{Q} = \mathbf{V} \frac{\Theta}{2}$ it holds [16]

$$\exp\left(\mathbf{V} \frac{\Theta}{2}\right) = \left[\cos\left(\frac{\Theta}{2}\right), \sin\left(\frac{\Theta}{2}\right) \mathbf{V} \right], \quad (45)$$

the inverse function of \exp for quaternions of the form (37) is then given by

$$\log : \mathbb{D}\mathbb{H}_1 \rightarrow T_1 \mathbb{D}\mathbb{H}_1 \quad (46)$$

$$\left[\cos\left(\frac{\Theta}{2}\right), \sin\left(\frac{\Theta}{2}\right) \mathbf{V} \right] \mapsto \mathbf{V} \frac{\Theta}{2}. \quad (47)$$

4. Pose Filter

With the ideas from Section 3 it is now possible to linearize the pose spaces $SO(3)$ and $SE(3)$ locally with the logarithm maps (24) and (47) using parallel transport (17) such

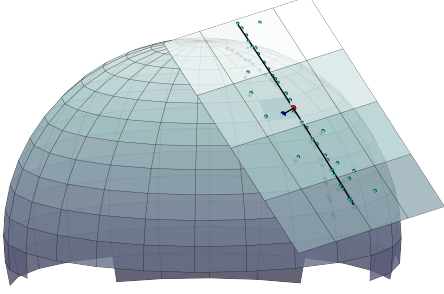


Figure 3. Robust PC-regression on tangent space. The figure illustrates the conceptual idea in 3D, rather than 8D.

that Euclidean methods can be applied for investigations and processing of pose data. A solution of an operation in tangent space $T_x\mathbb{G}$ can be mapped back thereafter with the exponential maps (22), (27) and (16) without affecting the Riemannian geometry of the manifold.

In the following, we develop a robust smoothing method in (dual) quaternion space by using the aforementioned operations to filter the pose data with a linear approximation of the rigid body movement in the tangent space.

4.1. Robust Motion Stabilization

A time-varying, sequential motion can be interpreted on the pose space \mathbb{DH}_1 as a high dimensional trajectory. Generally, due to lack of constraints on physical motion or tracking errors, this trajectory is highly non-linear, non-Gaussian and includes outliers. Moreover, it is non-uniformly sampled because of velocity changes or sudden jumps. We take this non-ideal setting into account and propose a parameter-free path smoothing algorithm which is robust, flexible, intuitive and could naturally benefit from the availability of the uncertainties in the pose estimation. Locally, we treat the trajectory as a linear one, and seek to find the linear association of the data points \mathbf{X} to the responses \mathbf{Y} such that $\mathbf{Y} = \mathbf{X}\beta + \epsilon$. A common way to discover this local relationship is using the Generalized Least Squares minimization

$$\beta^* = \arg \min_{\beta} (\mathbf{Y} - \mathbf{X}\beta)^T \mathbf{W} (\mathbf{Y} - \mathbf{X}\beta). \quad (48)$$

with diagonal weight matrix \mathbf{W} . The solution is given by weighted least squares regression

$$\beta^* = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{Y}. \quad (49)$$

Equivalently, when the data is rather unevenly distributed, one likes to prevent distinguishing the predictor and response, thereby rather looking for meaningful linear projections (e.g. maximum variance). Thus, such an hyperline can be obtained from the first principal component of the data points \mathbf{X} . Let $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ denote the singular value decomposition (SVD) of \mathbf{X} , with right singular vectors \mathbf{V} . Then, $\mathbf{V}\Delta\mathbf{V}^T$ gives the spectral decomposition of

Algorithm 1 irls_wpca : IRLS for weighted PCA.

Require: Local set of poses $\mathbf{X} = \{\mathbf{X}_i\}$, # Iterations N , Prior weights $\mathbf{w}^0 = \{w_i\}$

Ensure: PCA line \mathbf{l} with projections \mathbf{X}_{fit}

$\mathbf{w} \leftarrow \mathbf{w}_0$

for $i = 1 : N$ **do**

$\{\mathbf{X}_{proj}, \mathbf{l}\} \leftarrow \text{weighted_pca}(\mathbf{X}, \mathbf{w})$

Update \mathbf{w} using (52)

$\mathbf{w} \leftarrow \mathbf{w} \cdot \mathbf{w}^0 / \|\mathbf{w} \cdot \mathbf{w}^0\|$

\triangleright Dampen the estimates.

end for

Algorithm 2 Manifold PC-Local Regression.

Require: Set of poses $\mathbf{X} = \{\mathbf{X}_i\}$, Kernel size K , Prior weights $\mathbf{w}^0 = \{w_i\}$ for local window

Ensure: Filtered poses $\mathbf{X}^f = \{\mathbf{X}_i^f\}$

$\mathbf{X}^f \leftarrow []$

for $\mathbf{x}_i \in \mathbf{X}$ **do**

$\mathbf{X}_{\Omega} \leftarrow \{\mathbf{x}_k\} \in \Omega_i$

$\mathbf{X}_{\Omega}^t \leftarrow \log_{\mathbf{x}_i}(\mathbf{X}_{\Omega})$

$\mathbf{X}_{\Omega}^{proj} \leftarrow \text{irls_wpca}(\mathbf{X}_{\Omega}^t, \mathbf{w}_0)$

$\mathbf{x}_i^f \leftarrow \exp_{\mathbf{x}_i}(\mathbf{X}_{\Omega}^{proj}(i))$

$\mathbf{X}^f \leftarrow \mathbf{X}^f \cup \mathbf{x}_i^f$

end for

the covariance matrix $\mathbf{X}\mathbf{X}^T$ with the non-negative eigenvalues $\Delta = \text{diag}(\lambda_1 \dots \lambda_p)$. Weights are then transferred directly to the covariance matrix by

$$\mathbf{C} = \frac{1}{2\|\text{diag}(\mathbf{W})\|} (\mathbf{X} - \boldsymbol{\mu}) \mathbf{W} (\mathbf{X} - \boldsymbol{\mu})^T. \quad (50)$$

The columns of \mathbf{V} give an orthonormal set of eigenvectors and $\mathbf{X}\mathbf{v}_j$, the j^{th} principal component. The data can then be projected onto the first principal subspace resulting in the principal covariates $\mathbf{X}\mathbf{V}_k := \{\mathbf{X}\mathbf{v}_1 \dots \mathbf{X}\mathbf{v}_k\}$. To smooth the trajectory, the central point \mathbf{c} is projected onto the PC-line as illustrated on the plane in Figure 3. Note that this fit assumes a local Gaussian distribution, while the global distribution can be arbitrary.

While such PCA scheme holds for the Euclidean spaces, it does not generalize to arbitrary manifolds such as the dual quaternion space, because these spaces are not necessarily Euclidean. Buss and Fillmore [6] show that even regressing a great arc on the quaternion hypersphere has ambiguities. For the case of local regression, the central point of fitting is known which enables us to map the immediate neighbourhood onto the tangent space $T_x\mathbb{G}$, thereby circumventing the non-Euclidean nature of the dual quaternions. Thanks to the manifold structure, the tangent space locally behaves Euclidean and we can perform the fit. To smooth the curve, the central point is projected onto the regressed 8D-line and mapped back onto the manifold using the exponential map. We refer to this method as plain PCA filtering. Naturally, the 8D data points, which are closer to the center of the

local model \mathbf{c} are more relevant for the fit, as the linearity assumption decreases with the distance. Therefore we multiply each data point by a Gaussian prior function to down-weight the points based on their relative position:

$$w_i^0 = \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{c})^T \mathbf{D}(\mathbf{x}_i - \mathbf{c})\right), \quad (51)$$

where \mathbf{D} is a positive semi-definite distance metric explaining the region of influence. In the following experiments this approach is named weighted PCA (wPCA).

Oftentimes the pose space contains outliers, for which a naive fit does not work. To care for this, we introduce a re-weighted procedure, in which the residuals of the current fit are used to update the weights for the next iteration. This is commonly referred as iteratively reweighted least squares (IRLS). We use a simple distance based weight update:

$$\mathbf{w}_{i+1} = 1/\max\left(\delta, \frac{1}{K} \sum_{k=1}^K |\mathbf{r}_i^k|\right), \quad (52)$$

where δ is a small number, preventing division by zero and $\{\mathbf{r}_i^k\}$ are the residuals at iteration i .

Algorithm 2 summarizes our final implementation. As this method is independent of the structure of the parameter space, we can either use it individually on rotations and translations or on the dual quaternion space.

5. Experiments

In the following, the pose filtering methods presented in Section 4 are evaluated and compared to other approaches in two different scenarios. A first synthetic experiment analyzes the ability of the smoothing algorithms to recover a noisy pose series with outliers while the second evaluation is performed on a real dataset of natural hand movement in a collaborative medical robotic environment where tracking accuracy is crucial.

5.1. Synthetic Tests

Our first test evaluates the robustness and accuracy of the tangent space regressors. In order to evaluate these properties we generate a synthetic dataset from a ground truth rigid body movement.

A set of five points $\mathbf{v}_i \in \mathbb{R}^3$ together with five values $\theta_i \in [0, 2\pi]$ is chosen as query points representing the rotation axis and angle of the rotations \mathbf{R}_i . Five points $\mathbf{t}_i \in [0, 1]^3$ represent the translational component of the poses. A cubic spline interpolates both the axes and angles and with (4) and (12) we get a pose representations in $\mathbb{H}_1 \times \mathbb{R}^3$ and \mathbb{DH}_1 . As the space \mathbb{R}^3 is already Euclidean we can perform a pose filtering for the 6-DoF pose both in \mathbb{DH}_1 and $\mathbb{H}_1 \times \mathbb{R}^3$. For the latter, all methods are applied twice on both spaces independently.

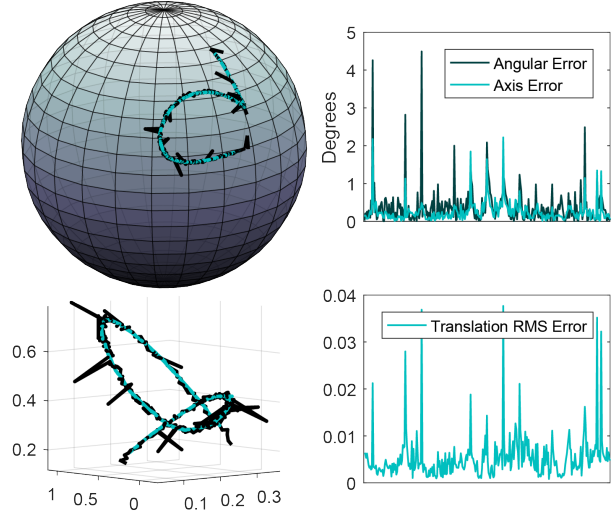


Figure 4. Dual IRLS applied to synthetic rigid body movement (black). On the left, the intersections of the rotation axis with S^2 are illustrated for both the noisy input (black) and the filtered pose (turquoise). The translation part is shown in the bottom. On the right, the angular and axis errors of the rotation are shown. The bottom plot shows the RMS error of the translational component.

To evaluate the performance of the local regression, we sample the ground truth pose series densely and apply additional noise uniformly distributed in $[-\sigma, \sigma]$ with $\sigma = 0.02$ to the angle and the axis of rotation as well as to the translation. On top, random outliers for 5% of the data points are created with an additional noise of $\sigma = 0.2$.

Then we run the methods PCA, wPCA, IRLS, Dual PCA, Dual wPCA, Dual IRLS as well as a Linear Kalman Filter on the data. We use a window size of 19 and the Kalman implementation [36] of MATLAB [22] with a covariance tuple of $[0.5, 2]$ for the rotation and $[0.2, 1]$ for the translation process noise and measurement noise covariance. The resulting pose set is illustrated together with the results for the Dual IRLS method in Figure 4. An error quantification for the different methods is given in Figure 5. For the Kalman filter, tradeoff values have been chosen which are still able to recover the pose without over-smoothing. However, the method only evaluates the past points and thus information of half the window size for future poses is not included which explains the performance difference. The direct local PCA methods perform equally well in the same error range while the weighting gain in the separate treatment is slightly better with 0.010 ± 0.020 in translation and $0.37 \pm 0.55^\circ$ rotation. The outlier aware IRLS method performs best for the angle while the improvement for the translation is with $2.1 \cdot 10^{-3}$ only minimal for the non-dual quaternions. It can be clearly seen that the treatment of outliers in the dual space \mathbb{DH}_1 helps to increase the accuracy to $4.3 \cdot 10^{-3}$ and 0.26° in the median. This can be intuitively understood through the fact that the local neighbourhood of

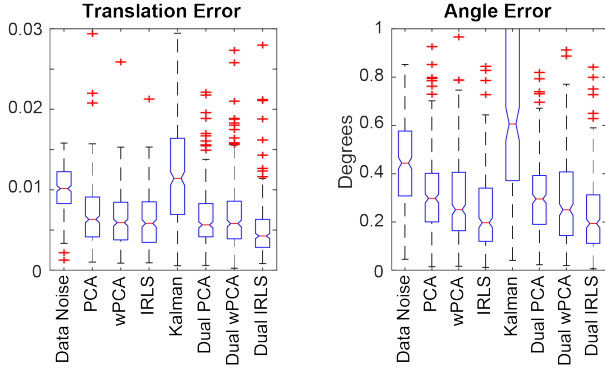


Figure 5. Box plots for performance tests of different pose filtering methods on synthetic data with the given noise shown in Data Noise. The illustrated Angle Error is the angular difference of the rotation axis to the ground truth.

the linear regression line in \mathbb{DH}_1 is much more restrictive than the joint neighbourhoods in $\mathbb{H}_1 \times \mathbb{R}^3$ where the effect of an outlier in the separate parameter spaces is higher.

It is also worth mentioning that the only parameter of the presented methods is the window size which directly reflects the movement speed of the displacements while the parameter adjustments for a filter method such as Kalman are more elaborate.

For visual comparison of the different methods on a synthetic pose stream with noise, please be referred to the supplementary video.¹

5.2. Tracking Stream Refinement

We compare our methods on the dataset of Busam et al. [5] where a robotic arm is set to gravity compensation mode with zero stiffness and a human operator performs a natural hand movement manipulating its end effector. The robot is tracked via a marker based stereo vision system running the tracking algorithm [4] and calibrated such that the forward kinematics of the industrial robotic manipulator provide ground truth poses with a precision of 0.05 mm.

The 30 Hz pose stream is fed into our filter pipelines and compared to the absolute poses of the dataset. Figure 6 illustrates the results where we use the same naming and parameters as in Section 5.1.

It is noteworthy that in this scenario, the wPCA methods perform best with $11.2 \pm 3.9\mu m$ (median $11.4\mu m$) and $0.7 \pm 1.4^\circ$ (median 0.5°) for the non-dual one while the IRLS methods give only mediocre results between PCA and wPCA both for quaternions ($13.0\mu m$, 0.6° median) and dual quaternions ($14.1\mu m$, 0.6° median). The Kalman filter again gives acceptable results for which heuristical parameter fine-tuning did not show any significant improvements. The advantage of the dual space robustification - which can

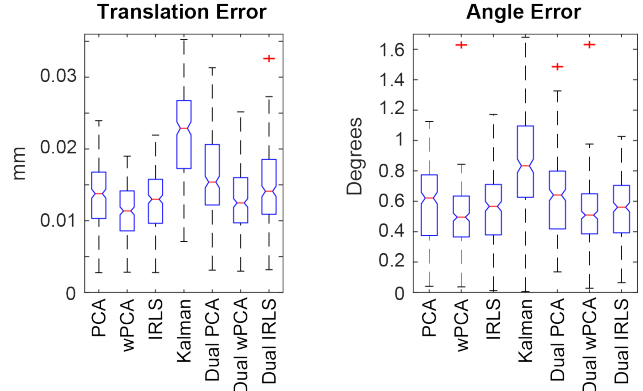


Figure 6. Box plots for tracking stream refinements of natural hand movement.

yield a significant improvement (see Section 5.1) in the case of outliers - is not applicable as there are only few outliers in the already quite accurate optical tracking data. The IRLS methods suffer from this problem as the weights for equally important data points are reduced. This results in case of reliable pose data in the fact that the separate treatment of translation and rotation is preferable as the non-dual regressors perform better.

6. Conclusion

We presented a method for camera pose filtering to recover smooth trajectories in pose space. The use of differential geometry helps to describe a local regression problem in the tangent space of unit (dual) quaternions. Applying a robust line fitting on the principal component of the pose measurements allows to filter an ordered pose series in a way that both the translation path as well as its orientation can be recovered. Besides smoothing and outlier-awareness, the method benefits from being non-parametric. In particular no explicit noise model needs to be modeled. Our experiments revealed that the dual space formulation robustifies the smoothing. As the logarithm and exponential maps are defined locally, the obvious downside of a local linearization is its need for a dense sampling around the touch point of the tangent space with the manifold. For an online application of the filtering also the moving window has to be cut to half or future points need to be extrapolated.

This work could be extended towards including an uncertainty measure into the local weights. An intuitive vision measure would be for example the backprojection error in a feature point tracking algorithm or the registration RMS of a 3D localization method. Moreover, the mentioned need for dense samples could be explicitly modeled including the timestamps in a tracking application. We believe that problems such as video stabilization or visual odometry could also benefit from the robust regression.

¹http://campar.in.tum.de/Chair/PublicationDetail?pub=busam2017_mvr3d

References

- [1] R. Ablamowicz and G. Sobczyk. *Lectures on Clifford (geometric) algebras and applications*. Springer Science & Business Media, 2004. 3
- [2] V. I. Arnol'd. The geometry of spherical curves and the algebra of quaternions. *Russian Mathematical Surveys*, 50(1):1, 1995. 2
- [3] C. Belta and V. Kumar. An svd-based projection method for interpolation on se (3). *IEEE transactions on Robotics and Automation*, 18(3):334–345, 2002. 1
- [4] B. Busam, M. Esposito, S. Che'Rose, N. Navab, and B. Frisch. A stereo vision approach for cooperative robotic movement therapy. In *Proceedings of the IEEE International Conference on Computer Vision Workshops, ICCVW*, pages 127–135, 2015. 8
- [5] B. Busam, M. Esposito, B. Frisch, and N. Navab. Quaternionic upsampling: Hyperspherical techniques for 6 dof pose tracking. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 629–638, Oct 2016. 1, 2, 8
- [6] S. R. Buss and J. P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics (TOG)*, 20(2):95–126, 2001. 6
- [7] H. H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 145–151, Jun 1991. 5
- [8] M. A. Clifford. Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, 1-4:381–395, 1871. 2
- [9] K. Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999. 4
- [10] M. Farenzena, A. Bartoli, and Y. Mezouar. Automatically smoothing camera pose using cross validation for sequential vision-based 3d mapping. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3616–3621. IEEE, 2008. 1, 2
- [11] O. D. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, fourth printing edition, 2001. 3
- [12] N. Filipe, M. Kontitsis, and P. Tsiotras. Extended kalman filter for spacecraft pose estimation using dual quaternions. *Journal of Guidance, Control, and Dynamics*, 38(9):1625–1641, 2015. 2
- [13] J. Gallier. Notes on differential geometry and lie groups. *University of Pennsylvania*, 2012. 4
- [14] W. R. Hamilton. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844. 2
- [15] C. Jia and B. L. Evans. Constrained 3d rotation smoothing via global manifold regression for video stabilization. *IEEE Trans. Signal Processing*, 62(13):3293–3304, 2014. 1, 2
- [16] L. Kavan, S. Collins, C. OSullivan, and J. Zara. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006-46*, 2006. 2, 5
- [17] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46. ACM, 2007. 1, 2
- [18] B. Kenwright. A beginners guide to dual-quaternions. *Winter School of Computer Graphics*, 2012. 2
- [19] B. Kenwright. Inverse kinematics with dual-quaternions, exponential-maps, and joint limits. *International Journal on Advances in Intelligent Systems*, 6(1 & 2), 2013. 3
- [20] Y. Kuang, A. Mao, G. Li, and Y. Xiong. A strategy of real-time animation of clothed body movement. In *International Conference on Multimedia Technology, ICMT*, pages 4793–4797, July 2011. 2
- [21] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005. 2, 3
- [22] MathWorks. Documentation of vision.kalmanfilter class., 2016. 7
- [23] R. Mukundan. Quaternions: From classical mechanics to computer graphics, and beyond. In *Proceedings of the 7th Asian Technology conference in Mathematics*, pages 97–105, 2002. 2
- [24] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994. 4
- [25] Y. Ng, B. Jiang, C. Yu, and H. Li. Non-iterative, fast se (3) path smoothing. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3172–3179. IEEE, 2016. 2
- [26] R. Parent. *Computer animation: Algorithms and Techniques*. Newnes, 2012. 2
- [27] E. Pennestrì and P. Valentini. Dual quaternions as a tool for rigid body motion analysis: A tutorial with an application to biomechanics. *Archive of Mechanical Engineering*, 57(2):187–205, 2010. 2
- [28] E. Pervin and J. A. Webb. Quaternions in computer vision and robotics. Technical report, Carnegie Mellon University. Computer Science Department, 1982. 2
- [29] J. Richter-Gebert and T. Orendt. *Geometrie-kalküle*, volume XI. Springer, 2009. 2
- [30] J. Selig. Exponential and cayley maps for dual quaternions. *Advances in applied Clifford algebras*, 20(3-4):923–936, 2010. 4
- [31] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985. 2
- [32] R. A. Srivatsan, G. T. Rosen, F. Naina, and H. Choset. Estimating se (3) elements using a dual-quaternion based linear kalman filter. *the proceedings of Robotics Science and Systems*, 2016. 2
- [33] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2441–2448. IEEE, 2011. 2
- [34] R. Tron, R. Vidal, and A. Terzis. Distributed pose averaging in camera networks via consensus on se (3). In *Distributed Smart Cameras, 2008. ICDCS 2008. Second ACM/IEEE International Conference on*, pages 1–10. IEEE, 2008. 3

- [35] M. Visser, S. Stramigioli, and C. Heemskerk. Cayley-hamilton for roboticists. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4187–4192. IEEE, 2006. [4](#)
- [36] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Department of Computer Science, 1995. [7](#)
- [37] J. Xu and K. H. Halse. Dual quaternion variational integrator for rigid body dynamic simulation. *arXiv preprint arXiv:1611.00616*, 2016. [1](#)