Quaternionic Upsampling: Hyperspherical Techniques for 6 DoF Pose Tracking

Benjamin Busam[1,2], Marco Esposito[1], Benjamin Frisch[1], Nassir Navab[1,3]

[1] Computer Aided Medical Procedures, Technische Universität München, Germany
[2] FRAMOS Imaging Sytems, Germany
[3] Computer Aided Medical Procedures, Johns Hopkins University, US

b.busam@framos.com, {marco.esposito, benjamin.frisch}@tum.de, navab@cs.tum.edu

Abstract

Fast real-time tracking is an integral component of modern 3D computer vision pipelines. Despite their advantages in accuracy and reliability, optical trackers suffer from limited acquisition rates depending either on intrinsic sensor capabilities or physical limitations such as exposure time. Moreover, data transmission and image processing produce latency in the pose stream.

We introduce quaternionic upsampling to overcome these problems. The technique models the pose parameters as points on multidimensional hyperspheres in (dual) quaternion space. In order to upsample the pose stream, we present several methods to sample points on geodesics and piecewise continuous curves on these manifolds and compare them regarding accuracy and computation efficiency.

With the unified approach of quaternionic upsampling, both interpolation and extrapolation in pose space can be done by continuous linear variation of only one sampling parameter. Since the method can be implemented rather efficiently, pose rates of over 4 *kHz and future pose predictions with an accuracy of* 128 μm *and* $0.5°$ *are possible in real-time. The method does not depend on a special tracking algorithm and can thus be used for any arbitrary* 3 *DoF or* 6 *DoF rotation or pose tracking system.*

1. Introduction

The possible loss of information about the 6 DoF position of an object because of delayed or interrupted data streams is a significant limitation in applications that use tracking systems or robotic manipulators [17]. The use of accurate predictions by extrapolating tracking data can be used to mitigate the effects of short-time information loss. In scenarios with multiple tracking systems or manipulators, upsampling can be used to synchronize movements or to align objects by recalculating time stamp based poses.

For instance, real-time collaborative robotics [10] require the online alignment of multiple coordinate systems. In dead reckoning, missing pose data can be calculated by extrapolation and the pose acquisition rate can be mathematically increased by physically motivated interpolation. This is particularly relevant for hybrid systems that combine update rates on different time scales such as from optical and inertial data streams [13].

This work proposes a thorough framework, based on quaternion representation, which is capable of prediction for both inter-point poses in the past and extra-point poses in the future for both rotation and translation components. To the best of our knowledge, there is no work using concepts from differential geometry and physical motivations for such interpolation and extrapolation of full 6 DoF pose measurements.

2. Related Literature

Hamilton's quaternions $\mathbb{H}$ [16] and - as their dual extension - the Clifford Algebra of dual quaternions [6] have a broad history in various fields of pure and applied geometry [2] [33], physics [38], robotics [31], and 3D computer graphics [32]. They are an integral part of modern real-time 3D processing software pipelines [27].

One of the main advantages of using quaternions to represent rotations is that unwanted effects such as gimbal lock are avoided [25]. Despite their usefulness for many real-time tasks, dual approaches did not receive the same attention as $\mathbb{H}$, that has been studied extensively. Kuang [22] showed the advantages of dual quaternion representations for animations with real-time motion calculations for clothed body movements. For visualization purposes, dual methods are used in blending [30], 3D pose transformations and complex hierarchical rigid body transforms [21]. Further applications of rotation parametrization with quaternions involve deformable model skinning [19],

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.71

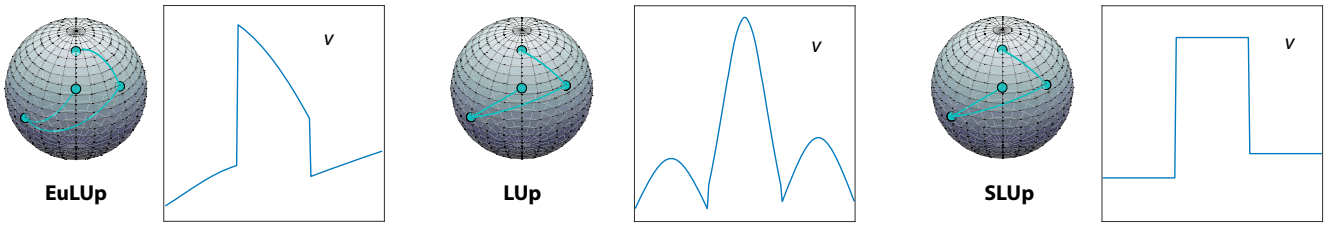

Figure 1. Resulting axis angle rotations of the upsampling methods EuLUp, LUp, SLUp and their angular velocity v.

mesh deformation [34] and sparse point cloud matching [37].

Physically or computationally motivated regression models for interpolation between key frames are essential to computer animation [29], where the start and end points of a transition or the object's trajectory are given and the information in between these two points has to be calculated. Quaternions seem well suited for this task due to an intuitive interpretation of special trajectories in $\mathbb{H}$. Shoemake [35] proposes a method for smooth interpolation of two rotations by motion along the geodesics on the 3-dimensional hypersphere S^3 with constant speed. Given the spherical linear interpolation character of this method, it is often referred to as SLERP. Barr [3] uses this work and includes velocity constraints for multi quaternion based spline interpolations. Later on, Nielson [28] used ν-quat splines including tension parameters as a tool for key-frame animation design. For a comparison of common interpolation techniques for rigid body rotations please refer to [7].

Instead of estimating unknown rotation parameters between known rotations, some authors focus on improving the supporting points for the trajectory fit. For virtual reality purposes, [24] compares different Kalman filter methods for denoising a quaternion tracking stream for human head and hand tracking and [39] describes a filter system for inertia-magnetic sensor data to track human limb segment orientation. There is only little literature on rotation extrapolation with quaternions. Chui [5] presents a way to construct a smooth trajectory for dead reckoning in virtual reality applications with special regard to oscillatory motions. The method is split in prediction and refinement of poses.

3. Mathematical Formulation

In the following, we review two concepts describing spatial displacement of rigid bodies. We first focus briefly on quaternions (3.1) and the corresponding algebra representing spatial rotations. Subsequently, we present their extension to dual quaternions in section 3.4.

3.1. Quaternions

Analogously to a complex number with an imaginary unit **i**, a **quaternion** can be defined with three imaginary units **i**, **j**, **k**.

Definition 1
*A **quaternion** $\boldsymbol{q}$ is an element of the algebra $\mathbb{H}$ that can be written as*

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

where $(q_1, q_2, q_3, q_4)^T \in \mathbb{R}^4$ is an element of $\mathbb{H}$ and the relation

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -\boldsymbol{1} \quad (2)$$

characterizes $\mathbb{H}$.

This definition forms a 4-dimensional normed division algebra over $\mathbb{R}$. We call $a = q_1 \in \mathbb{R}$ the scalar part and $\mathbf{v} = (q_2, q_3, q_4)^{\mathrm{T}} \in \mathbb{R}^3$ the vector part of the quaternion $\mathbf{q} := [a, \mathbf{v}]$.

We use quaternions with the standard arithmetic operations for addition, scalar multiplication and factorization. However, quaternion multiplication is not commutative in general as shown in the multiplication table for the imaginary units in Table 1. To use quaternions for spatial rotations one more ingredient is needed: an operator for conjugation.

·	**1**	**i**	**j**	**k**
1	**1**	**i**	**j**	**k**
i	**i**	**−1**	**k**	**−j**
j	**j**	**−k**	**−1**	**i**
k	**k**	**j**	**−i**	**−1**

Table 1. Multiplication table of $\mathbb{H}$

Definition 2
The conjugate $\bar{\boldsymbol{q}}$ of a quaternion $\boldsymbol{q} = q_1 + q_2\boldsymbol{i} + q_3\boldsymbol{j} + q_4\boldsymbol{k}$ is given by

$$\bar{\boldsymbol{q}} := q_1 - q_2\boldsymbol{i} - q_3\boldsymbol{j} - q_4\boldsymbol{k}. \quad (3)$$

3.2. Rotations with Quaternions

In quaternion notation, we describe a point $p = (x, y, z) \in \mathbb{R}^3$ by a **point quaternion** or pure quaternion

$$\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}. \quad (4)$$

It turns out [33] that a rotation of this point around an axis $\mathbf{u} = (u_1, u_2, u_3) \in \mathbb{R}^3$ can be written as

$$\mathbf{p}_{\text{new}} = \mathbf{q} \cdot \mathbf{p} \cdot \bar{\mathbf{q}} \tag{5}$$

with the **unit quaternion** $\mathbf{q} \in \mathbb{H}_1$

$$\mathbf{q} = a + u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k} =: [a, \mathbf{u}] \tag{6}$$

of length

$$1 \stackrel{!}{=} \|\mathbf{q}\| := \mathbf{q} \cdot \bar{\mathbf{q}} = a^2 + u_1^2 + u_2^2 + u_3^2. \tag{7}$$

Please note that one can also write a unit quaternion $\mathbf{r}$ as

$$\mathbf{r} = [\cos(\theta/2), \sin(\theta/2)\,\mathbf{v}] \tag{8}$$

which directly specifies the angle θ of rotation around the unit axis $\mathbf{v} = (v_1, v_2, v_3)$.
The choice of $\mathbb{H}_1$ is done to simplify the notation later and to ease the way to visualize a 4-dimensional quaternionic entity. However, any other sphere would work also. If we look at $\mathbf{q}^{-1} := \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2}$ which is the quaternion such that $\mathbf{q}\mathbf{q}^{-1} = \mathbf{1}$, we see that in $\mathbb{H}_1$ it is in particular $\mathbf{q}^{-1} = \bar{\mathbf{q}}$. To understand indeed that all quaternions on the same line through the origin represent the same rotation we look at

$$(\nu\mathbf{q}) \cdot \mathbf{p} \cdot (\nu\mathbf{q})^{-1} = \nu \cdot \mathbf{q} \cdot \mathbf{p} \cdot \mathbf{q}^{-1} \cdot \nu^{-1} \tag{9}$$

$$= \mathbf{q} \cdot \mathbf{p} \cdot \mathbf{q}^{-1} \cdot \nu\nu^{-1} = \mathbf{q} \cdot \mathbf{p} \cdot \mathbf{q}^{-1} \tag{10}$$

for an arbitrary quaternion $\mathbf{q} \in \mathbb{H} \setminus \{0\}$ with $\nu \in \mathbb{R} \setminus \{0\}$. This is a valuable information to describe the topology of the unit quaternion space.

3.3. Topology of $\mathbb{H}_1$

Due to its normalization, a unit quaternion has only three degrees of freedom. From a topological point of view, the unit quaternions form a 3-dimensional hypersphere $S^3 \in \mathbb{R}^4$ where two quaternions, namely $\mathbf{q}$ and $-\mathbf{q}$, represent the same rotation as we can see from equation 10. Thus, the unit quaternions form a double covering group of the group $SO(3)$ of 3D rotations. By identifying these antipodal points on S^3 we can construct the topology of $SO(3)$ which is then isomorphic to the real projective space $\mathbb{RP}^3$. From a perspective of geometric construction we can think of the hypersphere $\mathbb{H}_1$ in terms of $S^2 \times [0, 2\pi]$, where a resized S^2 sphere represents all rotations. A point on this sphere indicates the rotation axis and the radius r decodes the rotation angle. If we blow up a point at the origin - which represents the identity transform - the following spheres indicate rotations with increasing rotation angle $\theta = r \in [0, \pi]$ until radius $r = \pi$ is reached. From this point onwards, we shrink the radius r again while spherical points indicate rotations with angle $\theta = 2\pi - r$ until the sphere collapses in the origin where we started. A rotation with $\theta = 2\pi$ is identical with the identity transform. We use this more intuitive remapping of $\mathbb{H}_1$ as an idea for the visualization of rotations in our experiments.

For the further study of 3D rotations with quaternions it is crucial to understand this topological concepts since the metric structure of the unit quaternions only locally resembles $\mathbb{R}^3$. If we want to interpolate or extrapolate rotations we have to use trajectories on the hypersphere.

3.4. Dual Quaternions

Similar to the representation of rotations by quaternions of unit length, [11] we can use **dual quaternions** of unit length to represent rigid motions in space. These thoughts date back to Clifford [6], but one shall formulate this theory in a modern way.

A special, ordered set of real numbers that extends $\mathbb{R}$ can be defined by adding one new element ε. [9]

Definition 3
*A **dual number** z is an element of the algebra $\mathbb{D}$ that can be written as*

$$z = z_1 + z_2\varepsilon \tag{11}$$

where $z_1, z_2 \in \mathbb{R}$ and the relation

$$\varepsilon^2 = 0 \tag{12}$$

characterizes $\mathbb{D}$.

Combining this concept with quaternion algebra, we can define a **dual quaternion** as an ordered pair of quaternions with dual numbers as coefficients.

Definition 4
*A **dual quaternion** $\boldsymbol{Q} \in \mathbb{DH}$ is an ordered set of quaternions that can be written as*

$$\boldsymbol{Q} = \boldsymbol{r} + \boldsymbol{s}\varepsilon, \tag{13}$$

where $\boldsymbol{r}, \boldsymbol{s} \in \mathbb{H}$ with

$$\varepsilon^2 = 0, \quad \boldsymbol{i}\varepsilon = \varepsilon\boldsymbol{i}, \quad \boldsymbol{j}\varepsilon = \varepsilon\boldsymbol{j}, \quad \boldsymbol{k}\varepsilon = \varepsilon\boldsymbol{k}. \tag{14}$$

From the multiplication table 2 we notice that 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. Hence it is also non-commutative. Defining two conjugation operators on $\mathbb{DH}$ we get the following definition.

Definition 5
The conjugates $\bar{\boldsymbol{Q}}$ and $\hat{\boldsymbol{Q}}$ of a dual quaternion $\boldsymbol{Q} = \boldsymbol{r} + \boldsymbol{s}\varepsilon$ are given by

$$\bar{\boldsymbol{Q}} := \bar{\boldsymbol{r}} + \bar{\boldsymbol{s}}\varepsilon, \tag{15}$$

$$\hat{\boldsymbol{Q}} := \bar{\boldsymbol{r}} - \bar{\boldsymbol{s}}\varepsilon. \tag{16}$$

$\cdot$	$\mathbf{1}$	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$	ε	$\varepsilon\mathbf{i}$	$\varepsilon\mathbf{j}$	$\varepsilon\mathbf{k}$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$	ε	$\varepsilon\mathbf{i}$	$\varepsilon\mathbf{j}$	$\varepsilon\mathbf{k}$
$\mathbf{i}$	$\mathbf{i}$	$-\mathbf{1}$	$\mathbf{k}$	$-\mathbf{j}$	$\varepsilon\mathbf{i}$	$-\varepsilon$	$\varepsilon\mathbf{k}$	$-\varepsilon\mathbf{j}$
$\mathbf{j}$	$\mathbf{j}$	$-\mathbf{k}$	$-\mathbf{1}$	$\mathbf{i}$	$\varepsilon\mathbf{j}$	$-\varepsilon\mathbf{k}$	$-\varepsilon$	$\varepsilon\mathbf{i}$
$\mathbf{k}$	$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	$-\mathbf{1}$	$\varepsilon\mathbf{k}$	$\varepsilon\mathbf{j}$	$-\varepsilon\mathbf{i}$	$-\varepsilon$
ε	ε	$\varepsilon\mathbf{i}$	$\varepsilon\mathbf{j}$	$\varepsilon\mathbf{k}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
$\varepsilon\mathbf{i}$	$\varepsilon\mathbf{i}$	$-\varepsilon$	$\varepsilon\mathbf{k}$	$-\varepsilon\mathbf{j}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
$\varepsilon\mathbf{j}$	$\varepsilon\mathbf{j}$	$-\varepsilon\mathbf{k}$	$-\varepsilon$	$\varepsilon\mathbf{i}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
$\varepsilon\mathbf{k}$	$\varepsilon\mathbf{k}$	$\varepsilon\mathbf{j}$	$-\varepsilon\mathbf{i}$	$-\varepsilon$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$

Table 2. Multiplication table of $\mathbb{DH}$

3.5. Transformations with dual Quaternions

Similar to unit quaternions, we can represent a pose with a **unit dual quaternion** $\mathbf{Q} \in \mathbb{DH}_1$. Using induced standard arithmetic operations, we can calculate the constraints for such a dual quaternion.

$$1 \overset{!}{=} \|\mathbf{Q}\| := \mathbf{Q} \cdot \bar{\mathbf{Q}} \tag{17}$$
$$= (\mathbf{r} + \mathbf{s}\varepsilon) \cdot (\bar{\mathbf{r}} + \bar{\mathbf{s}}\varepsilon) \tag{18}$$
$$= \mathbf{r}\bar{\mathbf{r}} + (\mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}})\,\varepsilon. \tag{19}$$

This gives two distinct constraints on the dual quaternion which can be written as

$$\mathbf{r}\bar{\mathbf{r}} = 1 \quad \text{and} \quad \mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}} = 0. \tag{20}$$

Together, they reduce the eight parameters of a dual quaternion to the six degrees of freedom of a pose. We now use a unit dual quaternion to represent the rigid body displacement

$$X_{\text{new}} = \mathbf{R}X + t \tag{21}$$

for the point $X \in \mathbb{R}^3$ with a rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation $t \in \mathbb{R}^3$.

If we write the translation as a point quaternion $\mathbf{t}$ and the rotation as a unit quaternion $\mathbf{r}$, we construct the dual quaternion

$$\mathbf{Q} = \mathbf{r} + \frac{1}{2}\mathbf{t}\mathbf{r}\varepsilon. \tag{22}$$

Direct calculation of the dual quaternion norm $\|\mathbf{Q}\|$ shows that $\mathbf{Q} \in \mathbb{DH}_1$.

Analogously to the case of unit quaternions for rotation we want to represent a rigid body motion by dual quaternion multiplication. For this we take the point quaternion $\mathbf{p} = [0, \mathbf{u}]$ and transform it to the **dual point quaternion P**

$$\mathbf{P} = \mathbf{1} + \mathbf{u}\varepsilon \tag{23}$$

whose dual part represents the 3D coordinates. With this we can formulate the rigid body displacement in dual quaternion notation and see

$$\mathbf{P}_{\text{new}} = \mathbf{Q} \cdot \mathbf{P} \cdot \hat{\mathbf{Q}} \tag{24}$$
$$= \left(\mathbf{r} + \frac{1}{2}\mathbf{t}\mathbf{r}\varepsilon\right)(\mathbf{1} + \mathbf{u}\varepsilon)\left(\bar{\mathbf{r}} - \frac{1}{2}\bar{\mathbf{t}}\bar{\mathbf{r}}\varepsilon\right) \tag{25}$$
$$= \left(\mathbf{r} + \frac{1}{2}\mathbf{t}\mathbf{r}\varepsilon + \mathbf{r}\mathbf{u}\varepsilon\right)\left(\bar{\mathbf{r}} - \frac{1}{2}\bar{\mathbf{r}}\bar{\mathbf{t}}\varepsilon\right) \tag{26}$$
$$= \mathbf{r}\bar{\mathbf{r}} + \left(\frac{1}{2}\mathbf{t}\mathbf{r}\bar{\mathbf{r}} + \mathbf{r}\mathbf{u}\bar{\mathbf{r}} - \frac{1}{2}\mathbf{r}\bar{\mathbf{r}}\bar{\mathbf{t}}\right)\varepsilon \tag{27}$$
$$= 1 + (\mathbf{r}\mathbf{u}\bar{\mathbf{r}} + \mathbf{t})\,\varepsilon, \tag{28}$$

where the last step is due to the fact that $\bar{\mathbf{t}} = -\mathbf{t}$ for the point quaternion $\mathbf{t}$. The rotation $\mathbf{r}\mathbf{u}\bar{\mathbf{r}}$ is the known rotation of a point quaternion from 5. Thus the result represents the rigid body displacement in dual quaternion notation.

3.6. Topology of $\mathbb{DH}_1$

The unit dual quaternions are isomorphic to the group of rigid body displacement $SE(3)$ [1] where $\mathbf{Q}$ and $-\mathbf{Q}$ define the same motion. To understand the topology of the unit dual quaternion space, let us have a closer look at the two constraints from 20.
The first equation $\|\mathbf{r}\| = 1$ demands that the quaternion part $\mathbf{r}$ of $\mathbf{Q}$ has unit length, hence $\mathbf{r} \in \mathbb{H}_1$. From a geometric point of view, this gives a 7-dimensional space restricting the DoFs for unit dual quaternions to 7. With the identification of antipodal points, this forms the 7-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 $\mathbb{RP}^7$. From a topological point of view, interpolating in $\mathbb{DH}_1$ becomes constructing trajectories on this quadric.

4. Pose Upsampling

The essence of this section is to firstly define an upsampler in transformation space and use this concepts to formulate different methods of performing interpolation and extrapolation with various established pose parametrizations built upon the concepts of section 3. These are discussed and compared later.

Definition 6
*An **upsampler** $\gamma \in C^0$ from two transformations ϕ_1 and ϕ_2 of the pose parameter space $\mathbb{Q}$ is given by*

$$\gamma : \mathbb{Q} \times \mathbb{Q} \times \mathbb{R}_0^+ \to \mathbb{Q} \tag{29}$$
$$(\phi_1, \phi_2, \tau) \mapsto \gamma(\phi_1, \phi_2, \tau) \tag{30}$$

with

$$\gamma(\phi_1, \phi_2, 0) = \phi_1, \tag{31}$$
$$\gamma(\phi_1, \phi_2, 1) = \phi_2. \tag{32}$$

The interval $[0, \infty]$ for τ represents the sampling space where $[0, 1]$ is our observation range. The upsampler performs an interpolation for $\tau \in [0, 1]$ and extrapolates for $\tau > 1$. Usually tracking systems provide poses at a constant frequency and we therefore extrapolate mostly with $\tau \in (1, 2)$. However, due to occlusions, network lags and other disruptions, a longer dead-reckoning might be necessary. Altogether, the interpolation can then be seen as a refinement of the time-dependent information in between these two poses and the extrapolation states an estimate for the transformations beyond the observable pose states. The description of upsampling methods for the translation component of a pose breaks down to trajectory interpolation with supporting points. Besides linear interpolation there are many elaborate methods to preserve kinematic properties such as velocity continuity. The description of the rotation part is much more complex and we consider this first. We close this part by encapsulating both: We formulate the upsampling on unit dual quaternion space which gives an efficient formulation for pose refinements.

4.1. Euler Angles & Rotation Matrices

Before we consider quaternionic spaces we formulate an upsampling technique for Euler angles since it may seem convenient to sample in their parameter space. Given the parameter vector $\mathbf{a}_i = (\alpha_i^x \alpha_i^y \alpha_i^z)$ and two rotations $\mathbf{a}_1$, $\mathbf{a}_2$, the **Eu**ler angle **L**inear **Up**sampling of all three components is given by

$$\text{EuLUp}(\mathbf{a}_1, \mathbf{a}_2, \tau) := (1 - \tau)\,\mathbf{a}_1 + \tau \mathbf{a}_2. \tag{33}$$

The physical interpretation of such an upsampling can be counter intuitive since the trajectory of a rotated point does not have to take the shortest path as the first part of Figure 1 shows. Instead it takes a detour.

An alternative approach using element-wise linear upsampling in the space of rotation matrices would be

$$\text{MLUp}(\mathbf{M}_1, \mathbf{M}_2, \tau) := (1 - \tau)\,\mathbf{M}_1 + \tau \mathbf{M}_2, \tag{34}$$

where $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{3\times3}$ with the constraints $\mathbf{M}^{\mathrm{T}} = \mathbf{M}^{-1}$ and $\det(\mathbf{M}) = 1$. However, the interpolant $\mathbf{M}_\tau$ of MLUp is not necessarily an orthonormal matrix and thus an element of the rotational group $SO(3)$ anymore. A scaling of the object can in particular collapse the whole object to a single point.
For the obvious drawbacks of these methods we now focus on rotations in $\mathbb{H}_1$.

4.2. Quaternions

A direct normalized **L**inear **Up**sampling (LUp) with the rotations $\mathbf{q}_1$ and $\mathbf{q}_2$ in $\mathbb{H}$ is given by

$$\text{LUp}(\mathbf{q}_1, \mathbf{q}_2, \tau) := \frac{(1 - \tau)\,\mathbf{q}_1 + \tau \mathbf{q}_2}{\|(1 - \tau)\,\mathbf{q}_1 + \tau \mathbf{q}_2\|}. \tag{35}$$

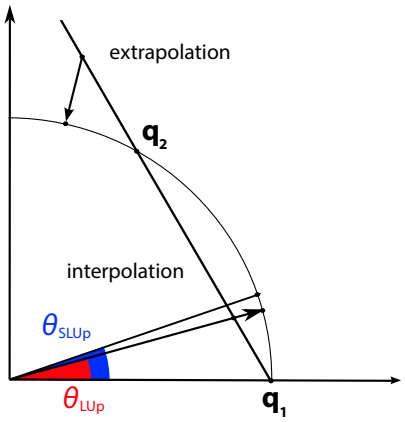

Figure 2. LUp and SLUp upsampling in $\mathbb{H}_1$

The interpolation trajectory is represented by the normalized straight line in $\mathbb{H}$ between the two quaternions. Even though $\mathbf{q}_i \in \mathbb{H}_1, i \in \{1, 2\}$, the trajectory takes the shortest path in $\mathbb{H}$ rather than $\mathbb{H}_1$ before normalization. We project the interpolated points onto the sphere $\mathbb{H}_1$ as illustrated in Figure 2, which gives the shortest path on $\mathbb{H}_1$. However, due to the projection, the sampling rate is different and the angular velocity varies with its maximum in the middle while continuously decreasing afterwards. From an efficiency point of view this method still provides a quick way for upsampling of rotations which is a good choice for critical real-time scenarios especially if many interpolations (rather than extrapolations) are needed.

With the ideas from section 3.3 we can operate directly on the geodesics of the hypersphere by moving at constant speed along its great arcs.
To be able to formulate a proper upsampling method on these geodesics, we introduce the induced operators

Definition 7
For $\boldsymbol{p} \in \mathbb{H}_1$, $\theta \in \mathbb{R}$, $\boldsymbol{v} \in \mathbb{R}^3$, $\|\boldsymbol{v}\| = 1$ *with* $\boldsymbol{p} = [0, \theta \boldsymbol{v}]$ *it holds*

$$\exp(\boldsymbol{p}) := [\cos(\theta), \sin(\theta)\,\boldsymbol{v}] =: \boldsymbol{r} \tag{36}$$

And with the inverse function $\log(\mathbf{r}) := [0, \theta \mathbf{v}]$ we can naturally define the exponentiation

$$\mathbf{q}^\tau := \exp(\tau \log(\theta \mathbf{q})) = \cos(\tau\theta) + \sin(\tau\theta)\,\mathbf{v}. \tag{37}$$

This gives the **S**pherical **L**inear **Up**sampling with $\mathbf{q}_1$ and $\mathbf{q}_2$ in $\mathbb{H}_1$ as

$$\text{SLUp}(\mathbf{q}_1, \mathbf{q}_2, \tau) := \mathbf{q}_1 \cdot (\bar{\mathbf{q}}_1 \cdot \mathbf{q}_2)^\tau. \tag{38}$$

Understanding that SLUp performs the upsampling on the great arcs of the quaternionic hypersphere by taking the shortest path from $\mathbf{q}_1$ to $\mathbf{q}_2$ is indeed a non-trivial task. The interested reader may be referred to Shoemake [35] who

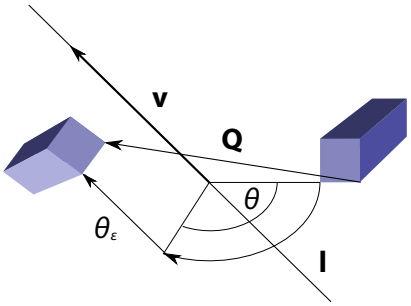

Figure 3. Screw linear displacement

first introduced the Spherical Linear Interpolation (SLERP) which is the basis for SLUp.

We will now also include the translational component and see that this concepts can be transferred to $\mathbb{DH}$, too.

4.3. Dual Quaternions

Besides the efficiency and compactness of dual quaternions [14] compared to matrix representation of displacements, we can use them for upsampling.
Let us start by introducing a shorthand for dual trigonometric operators

Definition 8
The dual operators for sine and cosine for $\Theta = \theta + \theta_\varepsilon \varepsilon \in \mathbb{DR}$ *are*

$$\sin(\Theta) := \sin(\theta) + \theta_\varepsilon \cos(\theta)\,\varepsilon \tag{39}$$
$$\cos(\Theta) := \cos(\theta) - \theta_\varepsilon \sin(\theta)\,\varepsilon. \tag{40}$$

It becomes handy that every unit dual quaternion $\mathbf{Q} \in \mathbb{DH}_1$ can be written [8] as

$$\mathbf{Q} = \mathbf{q} + \mathbf{q}_\varepsilon \varepsilon \tag{41}$$
$$= \cos\left(\frac{\Theta}{2}\right) + \sin\left(\frac{\Theta}{2}\right)\mathbf{V} \tag{42}$$

with the dual entities

$$\Theta = \theta + \theta_\varepsilon \varepsilon \in \mathbb{DR} \tag{43}$$
$$\mathbf{V} = \mathbf{v} + \mathbf{v}_\varepsilon \varepsilon \in \mathbb{DH}_1. \tag{44}$$

This is very similar to our observations in the quaternion case 8.
The dual vector $\mathbf{V}$ represents the axis $\mathbf{l}$ of a screw motion [18] with its direction vector $\mathbf{v}$ and moment $\mathbf{v}_\varepsilon$ (i.e. $\mathbf{v}_\varepsilon = \mathbf{p} \times \mathbf{v}\ \forall\ \mathbf{p} \in \mathbf{l}$). The dual angle Θ contains both the translation length θ_ε and the angle of rotation θ. This is visualized in Figure 3. Using the series definition of the exponential function adapted to $\mathbb{DH}_1$ we can formulate the exponential map on the 6-dimensional manifold of unit dual quaternions and note

$$\exp(\Theta\mathbf{V}) = \cos(\Theta) + \sin(\Theta)\,\mathbf{V} =: \mathbf{Q}. \tag{45}$$

With its inverse function $\log(\mathbf{Q}) := \Theta\mathbf{V}$ we are able to define the power of a dual quaternion as

$$\mathbf{Q}^\tau := \exp(\tau \log(\mathbf{Q})) = \cos(\tau\Theta) + \sin(\tau\Theta)\,\mathbf{V}. \tag{46}$$

Even though the mapping between $SE(3)$ and $\mathbb{DH}_1$ is one to two where the quaternions $\mathbf{Q}$ and $-\mathbf{Q}$ represent the same transformation this does not hold for powers. From a geometrical viewpoint this can be seen as the fact that there are two possible rotations around the screw axis that yield the same result, clockwise and counterclockwise.

The most intuitive upsampling methods extends SLUp directly to dual quaternion space and performs a screw linear upsampling **ScLUp**.
Let us sample between the two unit dual quaternions $\mathbf{Q}_1$ and $\mathbf{Q}_2$ in their standard form

$$\mathbf{Q}_1 = \mathbf{r}_1 + \frac{1}{2}\mathbf{t}_1\mathbf{r}_1\varepsilon \tag{47}$$
$$\mathbf{Q}_2 = \mathbf{r}_2 + \frac{1}{2}\mathbf{t}_2\mathbf{r}_2\varepsilon. \tag{48}$$

For the sampling point at $\tau \in [0,1]$ we write

$$\mathrm{ScLUp}(\mathbf{Q}_1, \mathbf{Q}_2, \tau) := \mathbf{Q}_1 \cdot \left(\mathbf{Q}_1^{-1} \cdot \mathbf{Q}_2\right)^\tau. \tag{49}$$

With the inverse displacement $\mathbf{Q}_1^{-1} \in \mathbb{DH}_1$ we can reformulate the variable part as

$$\left(\bar{\mathbf{Q}}_1 \cdot \mathbf{Q}_2\right)^\tau = \left(\left(\bar{\mathbf{r}}_1 + \frac{1}{2}\bar{\mathbf{r}}_1\bar{\mathbf{t}}_1\varepsilon\right) \cdot \left(\mathbf{r}_2 + \frac{1}{2}\mathbf{t}_2\mathbf{r}_2\varepsilon\right)\right)^\tau \tag{50}$$
$$= \left(\bar{\mathbf{r}}_1\mathbf{r}_2 + \frac{1}{2}\left(\bar{\mathbf{r}}_1\mathbf{t}_2\mathbf{r}_2 - \bar{\mathbf{r}}_1\mathbf{t}_1\mathbf{r}_2\right)\varepsilon\right)^\tau \tag{51}$$
$$= \left(\bar{\mathbf{r}}_1\mathbf{r}_2 + \frac{1}{2}\bar{\mathbf{r}}_1\left(\mathbf{t}_2 - \mathbf{t}_1\right)\mathbf{r}_2\varepsilon\right)^\tau \tag{52}$$

which gives the rigid transformation from $\mathbf{Q}_1$ to $\mathbf{Q}_2$. With definition 45, we observe that both the angle of rotation and the amount of translation from one pose to the other vary linearly with respect to τ while the axis stays constant. Thus ScLUp interpolates on the shortest possible path while the speed of rotation and translation is kept constant and further performs the screw motion for extrapolation.

An element in unit dual quaternion space $\mathbb{DH}_1$ can be viewed as a compact reparametrization of an element in unit quaternion space $\mathbb{H}_1$ together with a translation vector from $\mathbb{R}^3$. Thus, separate interpolation in quaternion space

and linear translation interpolation become a unified interpolation in $\mathbb{DH}_1$. This results in faster possible interpolation using ScLUp as we show in section 5.3. However, for real-time tasks at very high speed or limited computation resources an even more efficient solution may be needed. To achieve such a formulation we generalize the computationally efficient LUp to dual quaternions and create normalized **D**ual quaternion **L**inear **Up**sampling (**DLUp**) which is directly derived from the ideas of dual quaternion linear blending (DQLB) which is used for interpolations in 3D animation [12] and for pose refinement [18].

$$\mathrm{DLUp}\left(\mathbf{Q}_1, \mathbf{Q}_2, \tau\right) := \frac{\left(1-\tau\right)\mathbf{Q}_1 + \tau\mathbf{Q}_2}{\left\|\left(1-\tau\right)\mathbf{Q}_1 + \tau\mathbf{Q}_2\right\|}. \tag{53}$$

For most cases, **DLUp** is a close approximation of ScLUp, however the angular difference of the two methods can differ up to $8.15°$ for interpolations [20].

5. Experiments

5.1. Visualization Methods

To visualize the angular velocity v of an upsampled sequence $(\phi_n)_n$ in parameter space $\mathbb{Q}$ we use the centred average

$$v : \mathbb{Q} \to \mathbb{R} \tag{54}$$

$$\phi_i \mapsto v\left(\phi_i\right) := \frac{\left\|\phi_i - \phi_{i-1}\right\| + \left\|\phi_i - \phi_{i+1}\right\|}{2}. \tag{55}$$

Thus for a sequence of n samples, the values $v\left(\phi_0\right)$ and $v\left(\phi_n\right)$ are not defined.

For the 6 DoF pose, we visualize only the more interesting rotational part. Illustrating a quaternion $\mathbf{q} \in \mathbb{H}$ on paper is difficult since $\mathbb{H}$ is 4-dimensional and the paper is planar. We remember our topological considerations from section 3.3 and use a projection of the rotation axis onto S_2 and illustrate the rotation angle changes in a separate plot. Figure 1 shows the different rotational upsamplers on a dummy sequence of pose quaternions together with their angular velocities. We note in particular the constant angular velocity for the interpolation on the geodesics for SLUp as well as the projection dependent angular acceleration for LUp. For a comparison on the displacement of an objects and an example of MLUp, please be referred to the supplementary video [1].

5.2. Experimental Setup

We compared the method for the different upsamplers EuLUp, LUp, and SLUp for accuracy in an extrapolation scenario and performed efficiency tests for DLUp and ScLUp. For the $\mathbb{H}_1$ upsamplers with two supporting quaternions, we used linear upsampling for the translational part of the pose. Since the dual methods feature efficient reparametrizations for their primal partners with a linear translation upsampling we used them only for efficiency evaluations.

We tested the method with three different tracking systems. A commercial tracking system (Polaris, Northern Digital Inc., Waterloo, Canada) with 60 Hz is used as well as two prototypes using the tracking algorithm [4] on two 2 MPix cameras at 15 Hz and on two VGA cameras (all four SMARTEK Vision, Croatia) at 30 Hz respectively. The FRAMOS Application Framework (FRAMOS Imaging Systems, Germany) is used as a platform for the 2D image processing and 3D tracking. For all presented methods the computation is done on an Intel Core i7-4770K at 3.5 GHz. As ground truth, we use the forward kinematics of an industrial robotic manipulator (LWR4+, KUKA Gmbh, Augsburg, Germany) which offers a comparable baseline thanks to the high precision and sampling frequency of its encoders (0.05 mm and 1 kHz respectively) [23]. A combined marker for the three tracking systems is attached to the end effector of the manipulator. The robot is programmed to run in gravity compensation mode (zero stiffness) such that it can be directly manipulated by a human operator. The time stamps of all systems are synchronized with precision better than a millisecond to a ground-truth NTP server.

In order to bring the sampled positions into the same reference frame, a hand-eye calibration between the robot and each tracking system was performed. The Tsai-Lenz algorithm [36] in the eye-on-base variant was adopted for this purpose. The ViSP [26] implementation was used with 20 sampled points during each calibration procedure.
For the extrapolation tests our upsampler provides a live data stream of poses and pose estimations for different times in the future (5, 20, 100, 500 ms). Upsampling is done only with respect to the previous and the current pose at a constant frame rate, following the measurements from the tracking system. Whenever a new pose is received the procedure snaps to this current measurement and restarts the extrapolation procedure.

5.3. Validation

Figure 4 shows the accuracy evaluation with the native tracking streams and the upsampled versions. We note that our pose predictions are in the range of the robot accuracy for all trackers and estimates for 5, 20, and 100 ms. The best performing tracker for the 100 ms estimates is the 15 Hz tracker with $128 \pm 5.7\mu m$ and and $0.5 \pm 0.3°$ accuracy on SLUp. This is twice the accuracy of the 60 Hz system. We assume the reliability for the estimates comes from its more elaborate algorithmic pipeline [4] while the

[1] Video available for download at http://campar.in.tum.de/Chair/PublicationDetail?pub=busam2016_3dv.

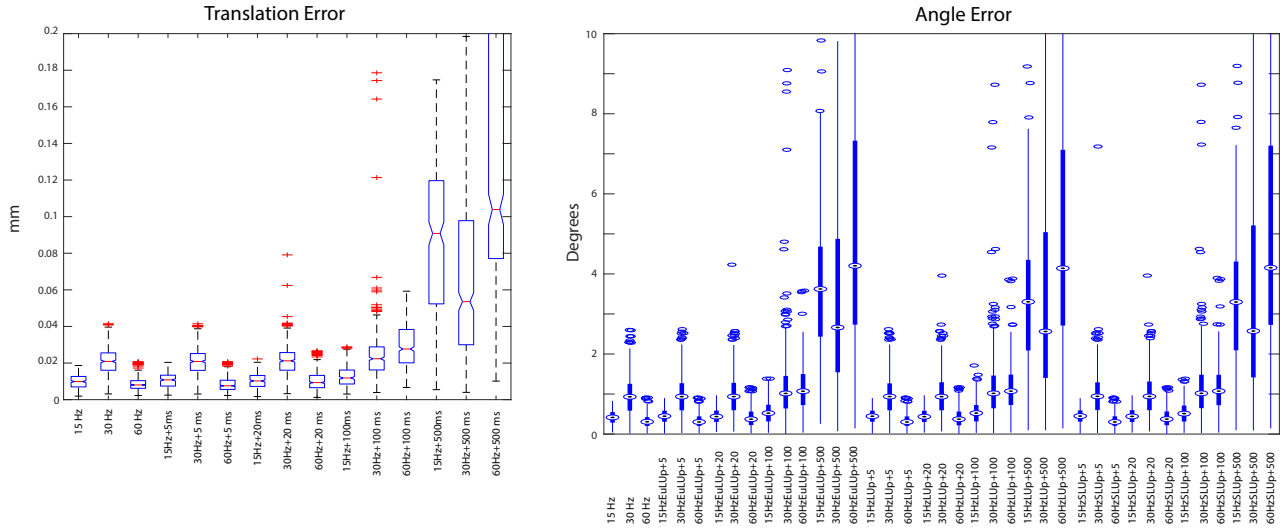

Figure 4. Performance tests of quaternionic upsampling on test data set of free hand movement. The extrapolation is performed with the proposed methods EuLUp, LUp / DLUp, and SLUp / ScLUp for estimations with 5, 20, 100, and 500 ms respectively. The absolute system error is shown first.

VGA cameras had a smaller overall mean error during stereo calibration (0.23 pixels vs. 0.49 pixels for the 30 Hz stereo tracker). The deviation for the 500 ms estimates can be explained by the major amplitude of the movement in that timespan. The proposed methods perform equally well with respect to the rotational error, although ScLUp has a physically more meaningful constant velocity graph.

For efficiency tests, all upsamplers are implemented using C++ together with the quaternion class of Eigen [15] which we enhanced for dual quaternion methods. The number of FLOPS for interpolation and extrapolation does not differ thus we evaluate the maximum pose sampling rate for interpolation between two fixed poses. The average results for the pose rate based on 10.000 calculations are summarized in Table 3. As we can see, splitting the methods in translation and rotation components (EuLUp, LUp, SLUp) has a negative impact on the upsampling performance. The fastest methods (DLUp, ScLUp) apply dual number theory and encapsulate the whole 6 DoF pose as one element on the hypersphere $\mathbb{DH}_1$. While the physically intuitive SLUp performs the slowest, its dual counterpart implementation ScLUp can be used for accurate high-speed online upsampling tasks.

EuLUp	LUp	SLUp	DLUp	ScLUp
469 Hz	306 kHz	236 Hz	4.55 kHz	2.53 kHz

Table 3. Efficiency Comparison of Upsampling Methods

6. Conclusion

We presented a unified approach for upsampling of 6 DoF and 3 DoF tracking data which is capable of interpolation and extrapolation by varying just one sampling parameter linearly. The method can be used to increase the measurement rate and robustness of any tracking system. Evaluation shows that an accurate tracker can give reliable estimates for future movement. We show that this method can predict human hand motion for up to 100 ms with an acceptable precision (0.1 mm, 0.5°). Obviously, steady movements increase the duration for which an acceptable prediction is possible.

This work could be extended towards including noise reduction of the incoming data stream and considering the pose history as a series of poses to preserve further differential geometric constraints or to minimize physical entities such as angular momentum. Finally, the combination with human movement recognition could introduce an intelligent movement prediction.

References

[1] R. Ablamowicz and G. Sobczyk. *Lectures on Clifford (geometric) algebras and applications*. Springer Science & Business Media, 2004. 4

[2] V. I. Arnol'd. The geometry of spherical curves and the algebra of quaternions. *Russian Mathematical Surveys*, 50(1):1, 1995. 1

[3] A. H. Barr, B. Currin, S. Gabriel, and J. F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 313–320, New York, NY, USA, 1992. ACM. 2

[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. 7

[5] Y.-P. Chui and P.-A. Heng. Adaptive attitude dead-reckoning by cumulative polynomial extrapolation of quaternions. In *Fifth IEEE International Workshop on Distributed Simulation and Real-Time Applications*, pages 45–52, Aug 2001. 2

[6] M. A. Clifford. Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, 1-4:381–395, 1871. 1, 3

[7] E. B. Dam, M. Koch, and M. Lillholm. *Quaternions, interpolation and animation*. Datalogisk Institut, Københavns Universitet, 1998. 2

[8] K. Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999. 6

[9] Z. Ercan and S. Yce. On properties of the dual quaternions. *European Journal of Pure and Applied Mathematics*, 4(2):142–146, 2011. 3

[10] M. Esposito, B. Busam, C. Hennersperger, J. Rackerseder, A. Lu, N. Navab, and B. Frisch. Cooperative robotic gamma imaging: Enhancing us-guided needle biopsy. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer Assisted Interventions*, MICCAI, October 2015. 1

[11] O. D. Faugeras. *Three-Dimensional Computer Vision.* The MIT Press, fourth printing edition, 2001. 3

[12] X. Feng and W. Wan. Dual quaternion blending algorithm and its application in character animation. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(10):5553–5562, 2013. 7

[13] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington. Flighttracker: A novel optical / inertial tracker for cockpit enhanced vision. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR, pages 212–221, 2004. 1

[14] J. Funda, R. H. Taylor, and R. P. Paul. On homogeneous transforms, quaternions, and computational efficiency. *IEEE Transactions on Robotics and Automation*, 6(3):382–388, June 1990. 6

[15] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 8

[16] 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. 1

[17] B. Johnson and G. Somu. Robotic telesurgery: Benefits beyond barriers. *BMH Medical Journal*, 3(2), 2016. 1

[18] L. Kavan, S. Collins, C. O'Sullivan, and J. Žára. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006-46*, 2006. 6, 7

[19] 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*, I3D, pages 39–46, New York, USA, 2007. ACM. 1

[20] L. Kavan and J. Žára. Spherical blend skinning: A real-time deformation of articulated models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, I3D '05, pages 9–16, New York, NY, USA, 2005. ACM. 7

[21] B. Kenwright. A beginners guide to dual-quaternions. *Winter School of Computer Graphics*, 2012. 1

[22] 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. 1

[23] KUKA GmbH. Kuka lwr. user-friendly, sensitive and flexible. http://www.kukaconnect.com/wp-content/uploads/2012/07/KUKA_LBR4plus_ENLISCH.pdf, 2012. 7

[24] J. J. LaViola. A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *Proceedings of the American Control Conference*, volume 3, pages 2435–2440, June 2003. 2

[25] 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. 1

[26] É. Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine*, 12(4):40–52, 2005. 7

[27] 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. 1

[28] G. M. Nielson. ν-quaternion splines for the smooth interpolation of orientations. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):224–229, March 2004. 2

[29] R. Parent. *Computer animation: Algorithms and Techniques*. Newnes, 2012. 2

[30] 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. 1

[31] E. Pervin and J. A. Webb. Quaternions in computer vision and robotics. Technical report, Carnegie Mellon University. Computer Science Department, 1982. 1

[32] D. Pletinckx. Quaternion calculus as a basic tool in computer graphics. *The Visual Computer*, 5(1):2–13, 1989. 1

[33] J. Richter-Gebert and T. Orendt. *Geometriekalküle*, volume XI. Springer, 2009. 1, 3

[34] J. A. Samareh. Application of quaternions for mesh deformation. Technical report, NASA, 2002. 2

[35] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985. 2, 5

[36] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989. 7

[37] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, 1991. 2

[38] A. Yefremov. Quaternions and biquaternions: Algebra, geometry and physical theories. *arXiv preprint math-ph/0501055*, 2005. 1

[39] X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, 22(6):1216–1227, 2006. 2