General Construction of Time-Domain Filters for Orientation Data

Jehee Lee and Sung Yong Shin

Abstract— **Capturing live motion has gained considerable attention in computer animation as an important motion generation technique. Canned motion data comprise both position and orientation components. Although a great deal of signal processing methods are available for manipulating position data, the majority of these methods cannot be generalized easily to orientation data due to the inherent non-linearity of the orientation space. In this paper, we present a new scheme that enables us to apply a filter mask (or a convolution filter) to orientation data. The key idea is to transform the orientation data into their analogues in a vector space, to apply a filter mask on them, and then to transform the results back to the orientation space. This scheme gives time-domain filters for orientation data that are computationally efficient and satisfy such important properties as** *coordinate-invariance*, *time-invariance*, **and** *symmetry*. **Experimental results indicate that our scheme is useful for various purposes including smoothing and sharpening.**

Keywords— **Orientation and Rotation, Unit quaternions, Motion signal processing, LTI filters, Convolution filters, Coordinate-invariance, Time-invariance.**

I. Introduction

Achieving realistic motions is a fundamental issue in computer animation. Capturing live motion has gained considerable attention as a promising motion generation techniques. The proliferation of highly detailed motion data has stimulated the development of tools to edit canned motion data which include both position and orientation components. A significant body of research is available for processing position data, but research on orientation data has recently been emerging. In particular, we note the efforts to generalize conventional signal processing and geometric techniques to manipulate orientation data [1], [2], [3], [5], [9], [10], [15], [16], [17], [18], [22], [24], [25], [26].

LTI (Linear Time-Invariant) filtering, which is equivalent to convolution filtering, is a fundamental technique in modern signal processing and has been used over a wide range of signal processing applications that include time- and space-domain filtering, approximating frequency-domain filtering, and multiresolution analysis [7], [11], [12], [21]. Given a vector-valued signal $\mathbf{p_i} \in \mathbb{R}^\mathbf{3}$ and a filter mask $(a_{-k}, \cdots, a_0, \cdots, a_k)$, LTI filtering is to sum the products between the mask coefficients and the sample values under the mask at a specific position on the signal. The i-th filter response is

$$\mathcal{F}(\mathbf{p}_i) = a_{-k}\mathbf{p}_{i-k} + \cdots + a_0\mathbf{p}_i + \cdots + a_k\mathbf{p}_{i+k}. \qquad (1)$$

A variety of methods have been investigated to apply a filter mask to orientation signals; however, many of those methods suffer from the lack of such important filter properties as *coordinate-invariance, time-invariance*, and *symmetry*. Our goal is to find a better analogue of Equation (1), that is, a coordinate-invariant, time-invariant, and symmetric filter for orientation signals. We expect our orientation filter to be efficient as well. We do not focus on designing a specific filter but propose a general scheme applicable to a large class of filter masks. Our interest lies in affine-invariant filter masks of which coefficients are summed up to one due to their wide applicability.

In this paper, we represent 3-dimensional orientations by unit quaternions. However, our theory can be applied not only to unit quaternions but also to any orientation representation that form a *Lie group* [4]. Unit quaternions as well as rotation matrices form a Lie group, for which exponential and logarithmic maps are defined.

In the following section, we review previous approaches that have motivated our work. In Section III, we give a brief introduction to LTI filters and orientation representation. In Section IV, we present our filtering scheme in detail. Examples of orientation filters are given in Section V. In Section VI, we discuss detailed implementation issues and illustrate relevant experimental results. In Section VII, we discuss previous approaches in comparison to our filter design scheme. Finally, we conclude this paper in Section VIII.

II. Background

In this section, we discuss previous approaches to draw the requirements of quaternion filters. The difficulty of filtering unit quaternion data stems from the non-linear nature of the unit quaternion space. Since the unit quaternion space is not closed under addition and scalar multiplication, the weighted sum of unit quaternion points is generally not a quaternion of unit length. One popular approach is to apply a mask to each component of unit quaternions separately and then to re-normalize the filter response to guarantee the unit magnitude [1]; however, re-normalization can incur side effects such as singularity and unexpected distortion as discussed in Section VII-A.

Several methods employing exponential and logarithmic maps have been introduced to avoid re-normalization. One is to transform quaternion points to the tangent space through logarithmic mapping, to apply a filter to the points in the tangent space, and then to transform the results back through exponentiation [8], [9]. Johnstone and Williams suggested a similar method that uses rational mapping between $\mathbb{S}^3$ and $\mathbb{R}^3$ [13]. The common idea behind these methods is to linearize the input signal by projecting unit quaternions into a vector space. However, it is well known that any global mapping between $\mathbb{S}^3$ and $\mathbb{R}^3$ has a singularity. The major drawback of these *global linearization* methods is that undesirable results are produced if the input signal is near to a singular point. Therefore, the filter response is dependent on the position in the reference frame. This observa-

The authors are with the Department of Electric Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), 373-1 Kusung-dong Yusung-gu, Taejon, Republic of Korea. The first author recently moved to the Robotics Institute at Carnegie Mellon University.
E-mail: jehee@cs.cmu.edu, syshin@jupiter.kaist.ac.kr

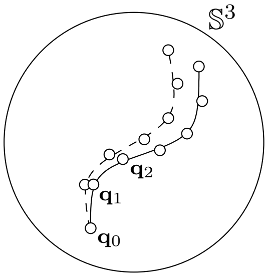


Fig. 1. The drifting problem in cumulative local linearization

tion implies that *global linearization* is not coordinate-invariant.

Fang *et al.* [5] suggested a *local linearization* method that is coordinate-invariant. They exploited spatial coherency to linearize the angular displacement between each pair of consecutive points in the signal. With this idea, they presented a three-step algorithm that first computes a sequence of angular displacements, then applies a filter to the sequence, and finally reconstructs a quaternion signal by integrating the filtered sequence of angular displacements. Even though coordinate-invariance is guaranteed by virtue of local linearization, the integration at the final step incurs a drifting problem; small changes of angular displacement vectors can result in large discrepancies at the end of the signal (Figure 1). This observation implies that local linearization is neither time-invariant nor symmetric.

Multi-linear methods based on *slerp* (spherical linear interpolation) are both coordinate-invariant and time-invariant [23], [25]. A typical example is the construction of quaternion Bézier curves suggested by Shoemake [25]. A point on a Bézier curve is defined as an affine combination of its control points, which can be computed by a series of linear interpolation by the de Casteljau algorithm. Shoemake generalized the de Casteljau algorithm to compute an affine combination of unit quaternions by converting it into a series of slerp. However, most of the variants in this category are computationally less efficient than the aforementioned linearization methods which need to evaluate exponential and logarithmic maps once per each sample point.

A stream of research addressed the problem from the viewpoint of functional optimization. Lee and Shin [17] formulated rotation smoothing as a non-linear optimization problem and derived smoothing operators from a series of fairness functionals defined on orientation data. Hsieh *et al.* [10], [15] presented a similar formulation for which the strain energy is minimized. They modified the traditional gradient-descent method to retain the unitariness of quaternions during optimization. The optimization approaches provide time-domain filters for smoothing orientation data. However, this idea is not based on filter masking and thus does not easily generalize to other types of filters.

Our goal is to develop a filtering scheme which is as efficient as the linearization methods and guarantees both coordinate- and time-invariance. Our approach can be classified as a local linearization method in the sense that it transforms the orientation data into their analogues in a vector space, applies a filter mask on them, and then transforms the results back to the orientation space. Unlike Fang *et al.*'s method [5], our filtering scheme has no drifting problem such as the one shown in Figure 1.

III. Preliminary

A. General Properties of LTI Filters

Applying filter masks to an input signal yields a class of time-domain filters. It is well known that those filters are both linear and time-invariant, and no others satisfy both of those properties [11], [12]. Hence, a filter in that class is called an LTI (Linear, Time-Invariant) filter. Let $\mathcal{F}$ be a filter that maps a discrete signal in $\mathbb{R}^d$ onto another in $\mathbb{R}^d$. The filter is *linear* if and only if

$$\mathcal{F}(a\mathbf{p}_i + b\mathbf{p}_i') = a\mathcal{F}(\mathbf{p}_i) + b\mathcal{F}(\mathbf{p}_i') \tag{2}$$

holds true for any given scalar values a and b, and vector-valued signals $\mathbf{p}_i, \mathbf{p}_i' \in \mathbb{R}^d$. The filter is *time-invariant* if its response does not depend on the time instance when the filter is applied. We can formulate this property more elegantly by introducing *shift operator* $\mathcal{S}^l$ that translates the signal in the time domain by l steps:

$$\mathcal{S}^l(\mathbf{p}_i) = \mathbf{p}_{i-l}. \tag{3}$$

Now, we can define the time-invariance as follows: The filter $\mathcal{F}$ is time-invariant if and only if $\mathcal{F}$ commutes with the shift operator, that is, $\mathcal{F} \circ \mathcal{S}^l = \mathcal{S}^l \circ \mathcal{F}$. Another important property is *symmetry*. The filter $\mathcal{F}$ is symmetric if $\mathcal{F}$ commutes with mirror reflection operator $\mathcal{R}$:

$$\mathcal{F} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{F}, \tag{4}$$

where $\mathcal{R}(\mathbf{p}_i) = \mathbf{p}_{-i}$. An LTI filter is symmetric if its mask coefficients are symmetric. Because asymmetric filters could shift the moments of a signal, symmetric filters are preferred in many cases [7], [11], [12].

Let $\Phi(\mathbf{p}_i) = A\mathbf{p}_i + \mathbf{b}$ be an affine transformation where A is a 3×3 matrix and $\mathbf{b}$ is a vector from $\mathbb{R}^3$. If the sum of its mask coefficients is one, an LTI filter $\mathcal{F}$ is invariant under affine coordinate transformation, that is,

$$\mathcal{F} \circ \Phi = \Phi \circ \mathcal{F}. \tag{5}$$

This equation implies that the filter response of $\mathcal{F}$ is independent of the choice of the coordinate system in which data points are represented. The affine coordinate transformation consists of translation, rotation, scaling, and shearing. Among them, only rotation is applicable to orientation data. Hence, in this paper, coordinate transformation means rotation and thus can be represented by a unit quaternion.

B. Orientation Representation

The four-dimensional space of quaternions is spanned by a real axis and three orthogonal imaginary axes, denoted by $\hat{\imath}, \hat{\jmath}$, and $\hat{k}$, which obey Hamilton's rules

$$\begin{array}{lll} \hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1, & & \\ \quad \hat{\imath}\hat{\jmath} = \hat{k}, & \hat{\jmath}\hat{\imath} = -\hat{k}, & \\ \quad \hat{\jmath}\hat{k} = \hat{\imath}, & \hat{k}\hat{\jmath} = -\hat{\imath}, & \text{and} \\ \quad \hat{k}\hat{\imath} = \hat{\jmath}, & \hat{\imath}\hat{k} = -\hat{\jmath}. & \end{array} \tag{6}$$

Clearly, quaternion multiplication is not commutative. Conventionally, we denote a quaternion $\mathbf{q} = w + x\hat{\imath} + y\hat{\jmath} + z\hat{k}$ by an ordered pair of a real number and a vector, $\mathbf{q} = (w, \mathbf{v}) \in \mathbb{R}\times\mathbb{R}^3$, where $\mathbf{v} = (x, y, z)$. Quaternions form a non-commutative group under multiplication with its identity $\mathbf{1} = (1, 0, 0, 0)$. The inverse of a quaternion $\mathbf{q}$ is $\mathbf{q}^{-1} = (w, -x, -y, -z)/(w^2+x^2+y^2+z^2)$. A quaternion of unit length is called a unit quaternion, which can be considered a point on the unit hyper-sphere $\mathbb{S}^3$.

From the fundamental theorem presented by Euler, any orientation of a rigid body can be represented as a rotation about a fixed axis $\hat{\mathbf{v}}$ by an angle $\theta \in [0, 2\pi)$, where $\hat{\mathbf{v}}$ is a 3-dimensional vector of unit length. Any rotation map $R_{\mathbf{q}} \in \mathbb{SO}(3)$ can be described by a unit quaternion $\mathbf{q} = (\cos\frac{\theta}{2}, \hat{\mathbf{v}}\sin\frac{\theta}{2}) \in \mathbb{S}^3$ as follows:

$$R_{\mathbf{q}}(\mathbf{p}) = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}, \quad \text{for } \mathbf{p} \in \mathbb{R}^3. \tag{7}$$

Here, the vector $\mathbf{p} = (x, y, z)$ is interpreted as a purely imaginary quaternion $(0, x, y, z)$. Note that $R_{\mathbf{q}} = R_{-\mathbf{q}}$; that is, two antipodal points, $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation in $\mathbb{SO}(3)$. Therefore, the mapping between $\mathbb{S}^3$ and $\mathbb{SO}(3)$ is two-to-one. Even with the two-to-one mapping, unit quaternions are used widely instead of rotation matrices in $\mathbb{SO}(3)$ for their compactness and computational efficiency.

The tangent vector at a point of a unit quaternion curve $\mathbf{q}(t) \in \mathbb{S}^3$ lies in the tangent space $T_{\mathbf{q}}\mathbb{S}^3$ at the point. Multiplying the tangent vector by the inverse of the quaternion point, the tangent space can be brought to coincide with the one at the identity $\mathbf{1} = (1, 0, 0, 0)$. Any tangent vector at the identity is perpendicular to the real axis and thus must be a purely imaginary quaternion, which corresponds to a vector in $\mathbb{R}^3$. In physics terminology, the purely imaginary tangent is identical to the angular velocity $\omega(t) \in \mathbb{R}^3$ of $\mathbf{q}(t)$:

$$\omega(t) = \frac{1}{2}\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t), \tag{8}$$

which is measured in the local coordinate frame specified by $\mathbf{q}(t)$. Here, $\frac{\omega(t)}{\|\omega(t)\|}$ is an instantaneous axis of the rotation, and $\|\omega(t)\|$ is the rate of change of the rotation angle about the axis. Because the unit quaternion space is folded by the antipodal equivalence, the angular velocity measured in $\mathbb{S}^3$ is twice as fast as the angular velocity measured in $\mathbb{SO}(3)$. The constant factor $\frac{1}{2}$ in Equation (8) keeps consistency between the unit quaternion space and the rotation space.

One of the main connections between vectors and unit quaternions is the exponential and logarithmic mapping. Given a purely imaginary quaternion $\mathbf{q} = (0, \mathbf{v})$, quaternion exponentiation gives a quaternion of unit length which can be expressed in a closed-form.

$$\exp(\mathbf{q}) = \exp(0, \mathbf{v}) = (\cos\|\mathbf{v}\|, \frac{\mathbf{v}}{\|\mathbf{v}\|}\sin\|\mathbf{v}\|) \in \mathbb{S}^3. \tag{9}$$

For simplicity, we often denote $\exp(0, \mathbf{v})$ as $\exp(\mathbf{v})$. This map has a singular point and is not one-to-one at that point. To define its inverse function, we limit the domain such that $\|\mathbf{v}\| < \pi$. Then, the exponential map becomes one-to-one and thus its inverse map $\log : \mathbb{S}^3\backslash(-1, 0, 0, 0) \rightarrow \mathbb{R}^3$ is well defined. Geometrically, exponentiation and logarithm give mappings between the tangent space $T_{\mathbf{1}}\mathbb{S}^3 \equiv \mathbb{R}^3$ at the identity and the unit quaternion space $\mathbb{S}^3$.

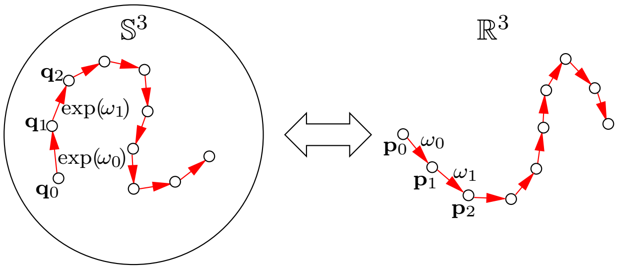


Fig. 2. The transformation between linear and angular signals

The *slerp* (spherical linear interpolation) introduced by Shoemake [25] parameterizes the points on a geodesic curve to compute the in-betweens of two given orientations $\mathbf{q}_1$ and $\mathbf{q}_2$.

$$\begin{aligned}\mathrm{slerp}_t(\mathbf{q}_1, \mathbf{q}_2) &= \mathbf{q}_1 \exp(t \cdot \log(\mathbf{q}_1^{-1}\mathbf{q}_2)) \\ &= \mathbf{q}_1(\mathbf{q}_1^{-1}\mathbf{q}_2)^t. \end{aligned} \tag{10}$$

$\mathrm{slerp}_t(\mathbf{q}_1, \mathbf{q}_2)$ describes an angular motion on the geodesic between $\mathbf{q}_1$ and $\mathbf{q}_2$ as t changes uniformly from zero to one. This motion has constant angular velocity $\log(\mathbf{q}_1^{-1}\mathbf{q}_2)$. In that sense, the *geodesic norm*

$$\mathrm{dist}(\mathbf{q}_1, \mathbf{p}_2) = \|\log(\mathbf{p}_1^{-1}\mathbf{p}_2)\| \tag{11}$$

is a natural distance metric in the unit quaternion space.

IV. Time-Domain Filters for Orientation Data

A. Key Idea

The ingredient of linearization is a transformation between the orientation space and a vector space that is used to project orientation data onto the vector space. To establish a relationship between a linear motion $\mathbf{p}(t) \in \mathbb{R}^3$ and an angular motion $\mathbf{q}(t) \in \mathbb{S}^3$, suppose that the linear velocity of $\mathbf{p}(t)$ and the angular velocity of $\mathbf{q}(t)$ have the same value. Then, we can consider their common velocity profile $\omega(t) \in \mathbb{R}^3$, which satisfies $\omega(t) = \dot{\mathbf{p}}(t) = \frac{1}{2}\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t)$ for any t. Note that both linear and angular velocities are measured in 3-dimensional vector spaces. If we integrate $\omega(t)$, the linear motion $\mathbf{p}(t)$ can be obtained as the solution of the differential equation $\omega(t) = \dot{\mathbf{p}}(t)$ with the initial condition $\mathbf{p}(0) = \mathbf{p}_0$. If we regard $\omega(t)$ as an angular velocity of a certain angular motion, then that motion $\mathbf{q}(t)$ can also be obtained from the differential equation $\omega(t) = \frac{1}{2}\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t)$ with the initial condition $\mathbf{q}(0) = \mathbf{q}_0$. This observation leads to a natural transformation between $\mathbf{p}(t)$ and $\mathbf{q}(t)$.

For a discrete velocity profile $\{\omega_j \in \mathbb{R}^3\}$, the discrete version of the differential equation $\omega(t) = \dot{\mathbf{p}}(t)$ gives a solution in a simple summation form $\mathbf{p}_i = \mathbf{p}_0 + \sum_{j=0}^{i-1}\omega_j$, where $\omega_j = \mathbf{p}_{j+1} - \mathbf{p}_j$. However, this kind of additive cumulation can be problematic for integrating unit quaternion data because of the unitariness constraint. Adding displacements to the unit quaternion data can result in deviation from the unit hypersphere $\mathbb{S}^3$. To avoid this problem, we employ exponential and logarithmic maps that give a “multiplicative” representation of angular displacements. Let $\{\mathbf{q}_i \in \mathbb{S}^3 | i \geq 0\}$ be a unit quaternion signal that forms a piecewise *slerp* curve. Through a simple derivation, each point $\mathbf{q}_i$ can be represented as a cumulation of displacements from a start point:

$$\begin{aligned}\mathbf{q}_i &= (\mathbf{q}_0\mathbf{q}_0^{-1})(\mathbf{q}_1\mathbf{q}_1^{-1})\cdots(\mathbf{q}_{i-1}\mathbf{q}_{i-1}^{-1})\mathbf{q}_i \\ &= \mathbf{q}_0(\mathbf{q}_0^{-1}\mathbf{q}_1)\cdots(\mathbf{q}_{i-1}^{-1}\mathbf{q}_i) \\ &= \mathbf{q}_0\prod_{j=0}^{i-1}\exp(\omega_j), \qquad (12)\end{aligned}$$

where $\omega_j = \log(\mathbf{q}_j^{-1}\mathbf{q}_{j+1}) \in \mathbb{R}^3$. Note that the angular displacement between two successive unit quaternions $\mathbf{q}_j$ and $\mathbf{q}_{j+1}$ can be parameterized by a 3-dimensional vector ω_j, which represents the angular velocity of the *slerp* motion that starts at $\mathbf{q}_j$ toward $\mathbf{q}_{j+1}$. Thus, we can reconstruct the original signal exactly from a start point $\mathbf{q}_0$ and a sequence of angular displacement vectors $(\omega_0, \omega_1, \omega_2, \cdots)$. It was proved that the multiplicative cumulation of angular displacements converges to the solution of the differential equation $\omega(t) = \frac{1}{2}\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t)$ as the time step between successive sample points approaches zero [14].

Given a discrete quaternion signal $Q = \{\mathbf{q}_i \in \mathbb{S}^3\}$, we define its vector counterpart $P = \{\mathbf{p}_i \in \mathbb{R}^3\}$ in such a way that each linear displacement $\mathbf{p}_{i+1} - \mathbf{p}_i$ equals the corresponding angular displacement $\log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})$. Let $p_0 \in \mathbb{S}^3$ and $q_0 \in \mathbb{R}^3$ be the start points of P and Q, respectively. Then, P and Q can be transformed to each other as follows (Figure 2):

$$\mathbf{p}_i = \mathbf{p}_0 + \sum_{j=0}^{i-1}\log(\mathbf{q}_j^{-1}\mathbf{q}_{j+1}), \quad \text{and} \qquad (13)$$

$$\mathbf{q}_i = \mathbf{q}_0\prod_{j=0}^{i-1}\exp(\mathbf{p}_{j+1} - \mathbf{p}_j). \qquad (14)$$

With this transformation, the estimated velocity of the linear motion represented by P is identical to the estimated angular velocity of the angular motion represented by Q.

B. Filter Design

As given in Equation (1), let $\mathcal{F}$ be an affine-invariant filter of which coefficients are summed up to one:

$$\mathcal{F}(\mathbf{p}_i) = a_{-k}\mathbf{p}_{i-k} + \cdots + a_0\mathbf{p}_i + \cdots + a_k\mathbf{p}_{i+k}, \qquad (15)$$

where $\sum_{m=-k}^{k} a_m = 1$. If we apply $\mathcal{F}$ to the vector signal P, then each point is displaced by $(\mathcal{F}(\mathbf{p}_i) - \mathbf{p}_i)$, which is the filter gain of $\mathcal{F}$. The key observation of our approach is that there exists a one-to-one correspondence between linear displacements (or linear velocity) and angular displacements (or angular velocity). We define the corresponding orientation filter $\mathcal{H}$ in such a way that it yields angular displacement $\log(\mathbf{q}_i^{-1}\mathcal{H}(\mathbf{q}_i))$, which is equal to the linear displacement $(\mathcal{F}(\mathbf{p}_i) - \mathbf{p}_i)$ gained by $\mathcal{F}$. The resulting orientation filter is

$$\mathcal{H}(\mathbf{q}_i) = \mathbf{q}_i\exp\Big(\mathcal{F}(\mathbf{p}_i) - \mathbf{p}_i\Big). \qquad (16)$$

The unitariness of filter responses is guaranteed because the unit quaternion space is closed under the quaternion multiplication. Conceptually, our filtering scheme is to transform the input orientation signal Q to its analogue P in a vector space, to apply a mask to P in a normal way, and finally to generate a filter response through the inverse transformation (Figure 3).

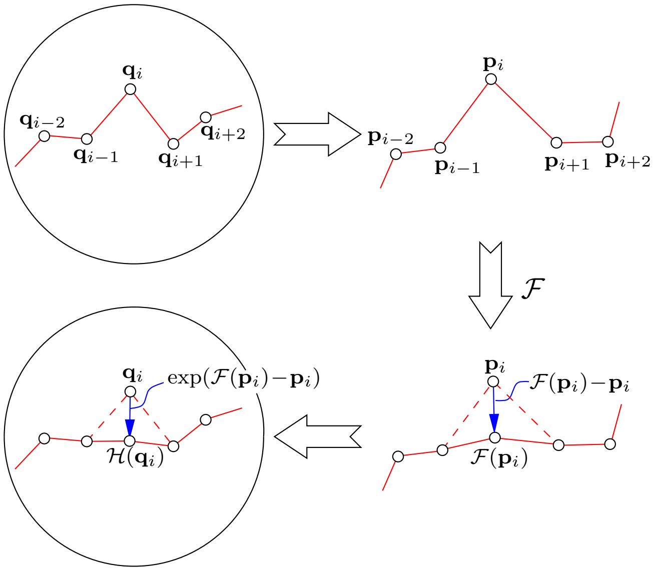


Fig. 3. The conceptual view of filtering orientation data

We can now expand and simplify Equation (16) step by step to reveal the algebraic and geometric structures of the filter. First, we need to show that our filter has a local support. The notion of local support is important in designing a filter that corresponds to a mask of finite size. Letting $\mathbf{p}_m^i = \mathbf{p}_{i+m} - \mathbf{p}_i$,

$$\begin{aligned}\mathcal{H}(\mathbf{q}_i) &= \mathbf{q}_i\exp\Big(\big(\sum_{m=-k}^{m=k} a_m\mathbf{p}_{i+m}\big) - \mathbf{p}_i\Big) \\ &= \mathbf{q}_i\exp\Big(\sum_{m=-k}^{m=k} a_m(\mathbf{p}_{i+m} - \mathbf{p}_i)\Big) \\ &= \mathbf{q}_i\exp\Big(\sum_{m=-k}^{m=k} a_m\mathbf{p}_m^i\Big). \qquad (17)\end{aligned}$$

In Equation (17), the angular displacement gained by $\mathcal{H}$ is described in terms of the masking operation over $\mathbf{p}_m^i$, which can be computed from Equation (13) as follows:

$$\mathbf{p}_m^i = \begin{cases}\sum_{j=1}^{m}\log(\mathbf{q}_{i+j-1}^{-1}\mathbf{q}_{i+j}), & \text{if } m \geq 1, \\ 0, & \text{if } m = 0, \\ \sum_{j=m}^{-1} -\log(\mathbf{q}_{i+j}^{-1}\mathbf{q}_{i+j+1}), & \text{if } m \leq -1.\end{cases} \qquad (18)$$

Clearly, $\mathcal{H}(\mathbf{q}_i)$ is locally supported by the neighboring points $(\mathbf{q}_{i-k}, \cdots, \mathbf{q}_i, \cdots, \mathbf{q}_{i+k})$. The size of support of $\mathcal{H}$ is identical to that of $\mathcal{F}$. One interesting observation is that the explicit evaluation of $\mathbf{p}_i$ and $\mathcal{F}(\mathbf{p}_i)$ is not actually needed for computing $\mathcal{H}(\mathbf{q}_i)$, although $\mathcal{H}$ is defined in terms of them.

Further simplification of Equation (17) will give an efficient algorithm to evaluate filter responses. Letting $\omega_i =$

$\log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})$,

$$\begin{aligned}
\mathcal{H}(\mathbf{q}_i) = \mathbf{q}_i \exp\Big(& a_1\omega_i \\
& + a_2(\omega_i + \omega_{i+1}) + \cdots \\
& + a_k(\omega_i + \omega_{i+1} + \cdots + \omega_{i+k-1}) \\
& - a_{-1}\omega_{i-1} \\
& - a_{-2}(\omega_{i-1} + \omega_{i-2}) - \cdots \\
& - a_{-k}(\omega_{i-1} + \omega_{i-2} + \cdots + \omega_{i-k})\Big) \\
= \mathbf{q}_i \exp\Big(& (a_1 + a_2 + \cdots + a_k)\omega_i \\
& + (a_2 + \cdots + a_k)\omega_{i+1} + \cdots \\
& + a_k\omega_{i+k-1} \\
& - (a_{-1} + a_{-2} + \cdots + a_{-k})\omega_{i-1} \\
& - (a_{-2} + \cdots + a_{-k})\omega_{i-2} - \cdots \\
= \mathbf{q}_i \exp\Big(& \sum_{m=-k}^{k-1} b_m\omega_{i+m}\Big),
\end{aligned} \tag{19}$$

where

$$b_m = \begin{cases} \sum_{j=m+1}^{k} a_j, & \text{if } 0 \le m \le k-1, \\ \sum_{j=-k}^{m} -a_j, & \text{if } -k \le m < 0. \end{cases} \tag{20}$$

Equation (19) lead to a filtering algorithm:

1. Compute $b_{-k}, \cdots, b_{k-1}$.
2. Compute $\omega_i = \log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})$ for all i.
3. Compute $\mathcal{H}(\mathbf{q}_i)$ for all i.

The most time-consuming operations in the algorithm are exponential and logarithmic maps that involve transcendence functions. Given n quaternion points, the second step of the algorithm computes logarithmic mapping n times if the input signal is cyclic, and $n-1$ times if the input signal is noncyclic. The third step computes exponential mapping n times. Therefore, we need to compute a pair of exponential and logarithmic maps for each point in the algorithm.

C. Properties of Orientation Filters

As mentioned earlier, the exponential map $\exp(\mathbf{v})$ is defined for all $\mathbf{v} \in \mathbb{R}^3$ but its inverse, that is the logarithm, is not well defined at $-\mathbf{I} = (-1, 0, 0, 0)$. To evaluate ω_i, we need to assume that the angle between any pair of consecutive points is smaller than π, that is, $\|\log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})\| < \pi$ for all i. π in the unit quaternion space is equivalent to 2π in the orientation space, and thus our assumption does not create a problem in practice.

The orientation filter $\mathcal{H}$ inherits important properties from LTI filters. The first property that we will prove in this section is coordinate-invariance. Due to this property, our filter yields the same results independent of the coordinate system in which the orientation data are represented. The following proposition shows that $\mathcal{H}$ is invariant under both local and global coordinate transformations, that is, $\mathcal{H}(\mathbf{q}_i)\mathbf{b} = \mathcal{H}(\mathbf{q}_i\mathbf{b})$ and $\mathbf{a}\mathcal{H}(\mathbf{q}_i) = \mathcal{H}(\mathbf{a}\mathbf{q}_i)$, respectively.

Proposition 1: $\mathcal{H}$ is coordinate-invariant, that is, $\mathbf{a}\mathcal{H}(\mathbf{q}_i)\mathbf{b} = \mathcal{H}(\mathbf{a}\mathbf{q}_i\mathbf{b})$ for any $\mathbf{a}$ and $\mathbf{b} \in \mathbb{S}^3$.

Proof: The first step of our proof is to show that $\exp(\mathbf{b}^{-1}\mathbf{v}\mathbf{b}) = \mathbf{b}^{-1}\exp(\mathbf{v})\mathbf{b}$ for any $\mathbf{v} \in \mathbb{R}^3$ and $\mathbf{b} \in \mathbb{S}^3$. Since $\|\mathbf{v}\| = \|\mathbf{b}^{-1}\mathbf{v}\mathbf{b}\|$,

$$\begin{aligned}
\exp(\mathbf{b}^{-1}\mathbf{v}\mathbf{b}) &= (\cos\|\mathbf{v}\|, \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\mathbf{b}^{-1}\mathbf{v}\mathbf{b}) \\
&= (\cos\|\mathbf{v}\|, \mathbf{0}) + (0, \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\mathbf{b}^{-1}\mathbf{v}\mathbf{b}) \\
&= \mathbf{b}^{-1}(\cos\|\mathbf{v}\|, \mathbf{0})\mathbf{b} + \mathbf{b}^{-1}(0, \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\mathbf{v})\mathbf{b} \\
&= \mathbf{b}^{-1}(\cos\|\mathbf{v}\|, \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\mathbf{v})\mathbf{b} \\
&= \mathbf{b}^{-1}\exp(\mathbf{v})\mathbf{b}.
\end{aligned}$$

Similarly, we can show that $\log(\mathbf{b}^{-1}\mathbf{q}\mathbf{b}) = \mathbf{b}^{-1}\log(\mathbf{q})\mathbf{b}$ for any $\mathbf{q}$ and $\mathbf{b} \in \mathbb{S}^3$. Then, we have

$$\begin{aligned}
\mathcal{H}(\mathbf{a}\mathbf{q}_i\mathbf{b}) &= \mathbf{a}\mathbf{q}_i\mathbf{b}\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{b}^{-1}\mathbf{q}_{i+m}^{-1}\mathbf{a}^{-1}\mathbf{a}\mathbf{q}_{i+m+1}\mathbf{b})\Big) \\
&= \mathbf{a}\mathbf{q}_i\mathbf{b}\exp\Big(\mathbf{b}^{-1}\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{q}_{i+m+1})\mathbf{b}\Big) \\
&= \mathbf{a}\mathbf{q}_i\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{q}_{i+m+1})\Big)\mathbf{b} \\
&= \mathbf{a}\mathcal{H}(\mathbf{q}_i)\mathbf{b}.
\end{aligned}$$

■

Since the support of $\mathcal{H}$ is finite, we can show that $\mathcal{H}$ is time-invariant.

Proposition 2: $\mathcal{H}$ is time-invariant.

Proof: Using Equation (19), we show that $\mathcal{H}$ commutes with $\mathcal{S}^l$ for any l:

$$\begin{aligned}
\mathcal{S}^l \circ \mathcal{H}(\mathbf{q}_i) &= \mathbf{q}_{i-l}\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{(i+m)-l}^{-1}\mathbf{q}_{(i+m+1)-l})\Big) \\
&= \mathbf{q}_{i-l}\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{(i-l)+m}^{-1}\mathbf{q}_{(i-l)+m+1})\Big) \\
&= \mathcal{H}(\mathbf{q}_{i-l}) = \mathcal{H} \circ \mathcal{S}^l(\mathbf{q}_i).
\end{aligned}$$

■

Now, we show that $\mathcal{H}$ is symmetric for any given symmetric coefficients.

Proposition 3: $\mathcal{H}$ is symmetric, if its coefficients are symmetric.

Proof: The proof will be complete if we show that $\mathcal{H}$ commutes with $\mathcal{R}$. We first expand $\mathcal{R} \circ \mathcal{H}$ using Equation (19):

$$\begin{aligned}
\mathcal{R} \circ \mathcal{H}(\mathbf{q}_i) &= \mathbf{q}_{-i}\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{-i-m}^{-1}\mathbf{q}_{-i-m-1})\Big) \\
&= \mathbf{q}_{-i}\exp\Big(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{-i-m-1}^{-1}\mathbf{q}_{-i-m})^{-1}\Big) \\
&= \mathbf{q}_{-i}\exp\Big(\sum_{m=-k}^{k-1} -b_m \log(\mathbf{q}_{-i-m-1}^{-1}\mathbf{q}_{-i-m})\Big).
\end{aligned}$$

From Equation (20), $b_m = -b_{-m-1}$ when $a_m = a_{-m}$. Therefore, letting $n = -m-1$,

$$\begin{aligned}\mathcal{R} \circ \mathcal{H}(\mathbf{q}_i) &= \mathbf{q}_{-i} \exp\left(\sum_{n=-k}^{k-1} -b_{-n-1} \log(\mathbf{q}^{-1}_{-i+n} \mathbf{q}_{-i+n+1}) \right) \\ &= \mathbf{q}_{-i} \exp\left(\sum_{n=-k}^{k-1} b_n \log(\mathbf{q}^{-1}_{-i+n} \mathbf{q}_{-i+n+1}) \right) \\ &= \mathcal{H}(\mathbf{q}_{-i}) = \mathcal{H} \circ \mathcal{R}(\mathbf{q}_i).\end{aligned}$$

■

V. Examples

In this section, we provide examples of orientation filters that correspond to popular filter masks for smoothing and sharpening.

A. Smoothing

Our first example is a smoothing mask that is of practical use in signal processing. The smoothness measure $\int \|\mathbf{p}''(t)\|^2 dt$ is minimized if the corresponding Euler-Lagrange equation $\mathbf{p}''''(t) = 0$ holds true [6]. The discrete version of the Euler-Lagrange equation, $\Delta^4 \mathbf{p}_i = 0$, is obtained by replacing differential operators with forward divided difference operators. It is well known that an iterative scheme using a local update rule

$$\mathbf{p}_i \leftarrow \mathbf{p}_i - \lambda \Delta^4 \mathbf{p}_i \tag{21}$$

gradually adjusts the data points to approach the optimal solution. Here, λ is a damping factor that controls the rate of convergence. This update rule yields an affine-invariant mask $(\frac{-\lambda}{24}, \frac{4\lambda}{24}, \frac{24-6\lambda}{24}, \frac{4\lambda}{24}, \frac{-\lambda}{24})$, which is very popular in signal processing applications. The corresponding orientation filter can be derived from Equation (19) as follows:

$$\mathcal{H}^S(\mathbf{q}_i) = \mathbf{q}_i \exp\left(\frac{\lambda}{24} (\omega_{i-2} - 3\omega_{i-1} + 3\omega_i - \omega_{i+1}) \right). \tag{22}$$

B. Blurring

A more popular class of filters are derived from the binomial distribution [11]. Binomial coefficients give a low-pass filter that suppresses Gaussian noise and blurs the details of the signal. The coefficients of an odd-sized $(2k+1)$ binomial mask can be written:

$$B_i^{2k+1} = \frac{1}{2^{2k}} \frac{(2k)!}{(k-i)!(k+i)!}, \qquad -k \le i \le k. \tag{23}$$

For a large k, the binomial distribution closely approximates Gaussian distribution. Thus, binomial masks are often used for approximating the ideal Gaussian filter. In particular, the orientation filter which corresponds to $B^5 = (\frac{1}{16}, \frac{4}{16}, \frac{6}{16}, \frac{4}{16}, \frac{1}{16})$ is derived from Equation (19):

$$\mathcal{H}^B(\mathbf{q}_i) = \mathbf{q}_i \exp\left(\frac{1}{16} (-\omega_{i-2} - 5\omega_{i-1} + 5\omega_i + \omega_{i+1}) \right). \tag{24}$$

C. Sharpening

Our final example is a high-frequency boost filter. Sharpening has commonly been achieved through filter masking to reveal the detailed features of the signal more clearly [7]. Because the fine details of a signal are the primary contributors to the high-frequency components of a signal, high-frequency boosting sharpens the signal. The high-frequency components can be extracted by blurring the original signal and subtracting the blurred signal from the original one. The high-frequency components thus obtained are added to the original signal as follows:

$$\begin{aligned}\text{(high-boost)} &= \text{(original)} + \lambda\text{(high-pass)} \\ &= \text{(original)} + \lambda(\text{(original)} - \text{(blurred)}).\end{aligned}$$

If the binomial mask B^5 is used for blurring the original signal, then the coefficients $(\frac{-\lambda}{16}, \frac{-4\lambda}{16}, \frac{16+10\lambda}{16}, \frac{-4\lambda}{16}, \frac{-\lambda}{16})$ for sharpening are obtained. By substituting the coefficients in Equation (19), we have the orientation filter

$$\mathcal{H}^U(\mathbf{q}_i) = \mathbf{q}_i \exp\left(\frac{\lambda}{16} (\omega_{i-2} + 5\omega_{i-1} - 5\omega_i - \omega_{i+1}) \right) \tag{25}$$

that performs high-frequency boosting on orientation signals.

VI. Experiments

A. Implementation Issues

There exists inherent ambiguity in a unit quaternion signal due to antipodal equivalence. Because any unit quaternion point and its antipode represent the same orientation, the signs of quaternion points in a captured signal are often chosen arbitrarily. However, filter responses are quite dependent on the signs, and thus they must be corrected consistently. We determine the sign of each point in the signal so that the point is placed near its adjacent neighbors. To do so, we first fix the sign of the first point $\mathbf{q}_0$ and then replace $\mathbf{q}_i$ with $-\mathbf{q}_i$ sequentially for each $i > 0$, if the geodesic distance between $\mathbf{q}_{i-1}$ and $\mathbf{q}_i$ is larger than $\frac{\pi}{2}$.

In general, the input signal is neither infinite nor periodic. The signal has boundary points, and the left boundary seldom has anything to do with the right boundary. A periodic extension can be expected to have a discontinuity. The natural way to avoid this discontinuity is to reflect the signal at its endpoints to seamlessly extend the signal. Let $(\mathbf{q}_0, \cdots, \mathbf{q}_n)$ be a unit quaternion signal and $\omega_i = \log(\mathbf{q}_i^{-1} \mathbf{q}_{i+1})$, $0 \le i < n$, be the angular displacements of the signal. Then, the extension of the signal at both boundaries yields

$$\omega_i = \begin{cases} \omega_{-i}, & \text{if } i < 0, \\ \omega_{2n-i-2}, & \text{if } i \ge n. \end{cases} \tag{26}$$

B. Experimental Results

We first apply our orientation filter to synthetic motion data to visualize the effect of filtering as shown in Figure 4. The initial orientation data (top left) are sampled uniformly from a quaternion spline curve, and random noises are added to the samples through multiplying each sample $\mathbf{q}_i$ by the exponent e^{δ_i} of a random 3D vector $\|\delta_i\| < 0.2$. The random perturbation makes each sample point displaced in a random direction up to 0.2 radians. We apply smoothing filter $\mathcal{H}^S$ to the initial motion once

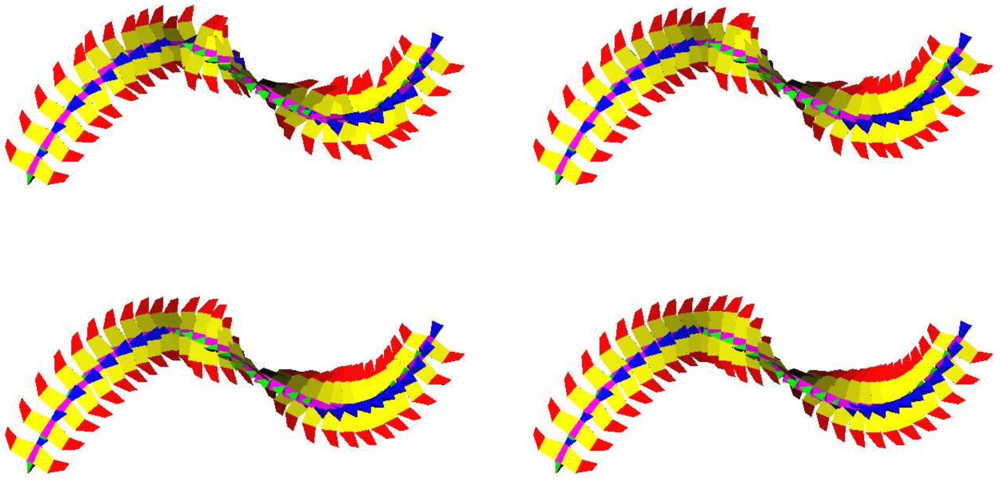


Fig. 4. Bird flying

(top right), twice (bottom left), and ten times (bottom right) to illustrate a series of incrementally refined results. The smoothing effect is clearly observed along the trajectories of the tips of the wings.

We also apply filters to captured motion data. Our motion capture system (*MotionStar*, Ascension Technology) consists of a magnetic field transmitter and 14 trackers, each of which is attached to a body segment of a puppeteer and detects both position and orientation of the segment measured in the global coordinate system. We captured the stretching motion of a live athletic sampled at the rate of 30 frames per second. Figure 5(a) shows the quaternion signal for the left shoulder. the X-axis represents the frame numbers of the signal and the Y-axis represents the value of each component of unit quaternions. The magnitude of angular acceleration is plotted to show the noise in the captured signal (Figure 5(b)).

In Figures 5(c) and (e), smoothing filters $\mathcal{H}^S$ and $\mathcal{H}^B$, respectively, were used to reduce the noise in the signal. Each filter was applied to the signal five times. The effect of smoothing is clearly shown in the corresponding magnitude plots of angular acceleration (Figures 5(d) and (f)). On the contrary, the high-frequency boosting filter $\mathcal{H}^U$ enhances the high-frequency components of the signal and thus the estimated angular acceleration vectors are magnified as expected (Figures 5(g) and (h)).

VII. Discussion

As mentioned earlier, several different methods for designing orientation filters are available. In this section, we compare our filter design scheme with those of others.

A. Re-normalization

In practice, the *re-normalization* method works well if the input signal is sufficiently dense. However, we have to deal with coarse signals in applications such as multiresolution analysis in which the input signal is filtered and down-sampled successively [3], [19], [20]. For example, if the average filter $(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ is applied to the signal that includes five consecutive sample points distributed evenly on a great arc, their average is zero and thus normalization is impossible (Figure 6). In this case, the filter response should be identical to the original point under the mask, since the average filter does not alter a symmetric configuration. In our filter, angular displacement vectors to be blended are cancelled out by symmetry and, therefore, the correct result is produced.

B. Global vs. Local Linearization

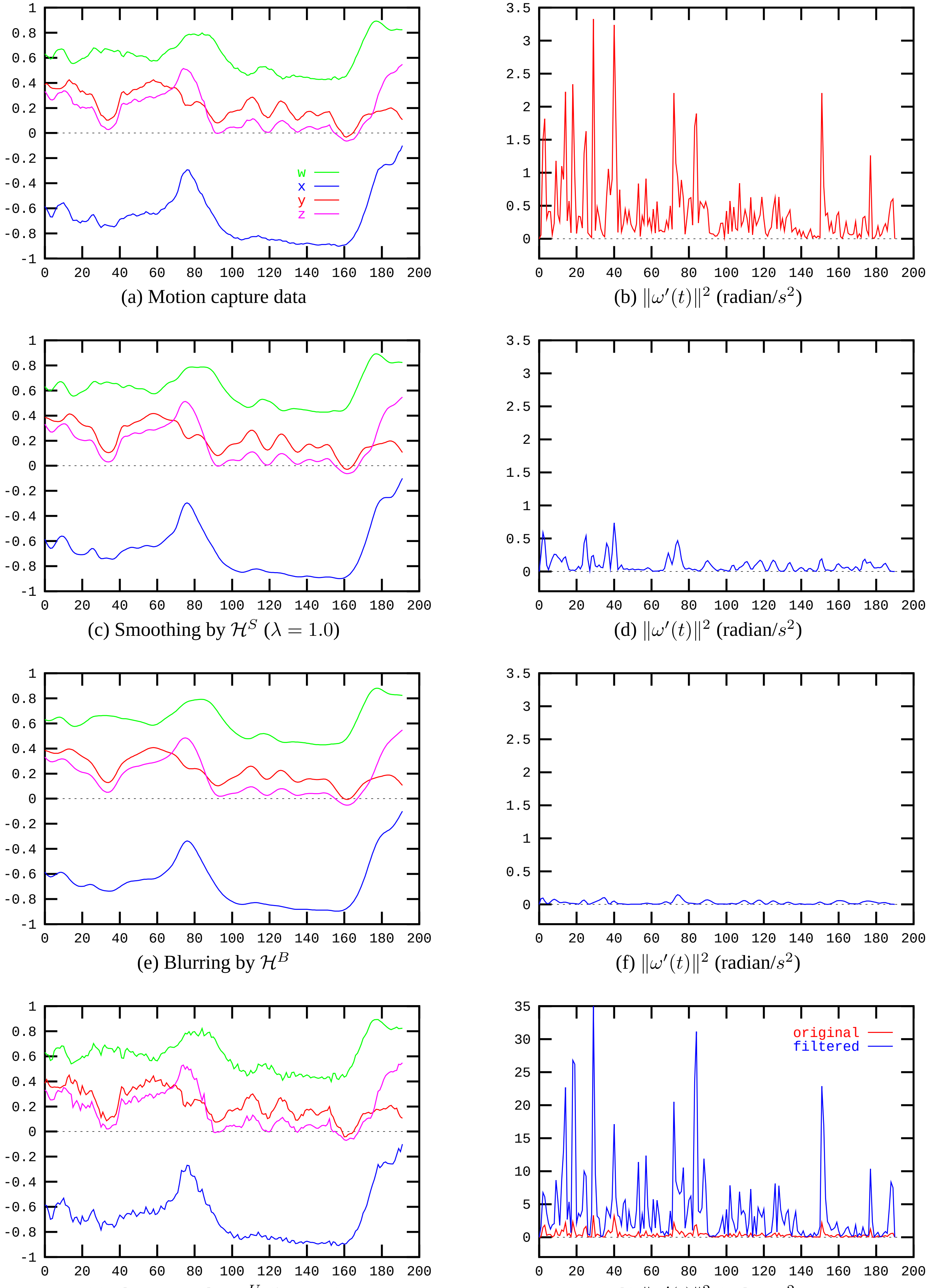
The naive use of the exponential and logarithm maps can lead to a troublesome orientation filter

$$\bar{\mathcal{H}}(\mathbf{q}_i) = \exp\Big(\mathcal{F}(\log(\mathbf{q}_i))\Big). \tag{27}$$

Because the logarithm of a unit quaternion is not well defined at $-I = (-1, 0, 0, 0)$, the filter $\bar{\mathcal{H}}$ is also undefined at that point. Furthermore, $\bar{\mathcal{H}}(\mathbf{q}_i)$ is not invariant even under constant rotation, and thus we can have quite different results for the change of the coordinate system in which the orientation data are represented. The main reason for these problems is that there is no globally non-singular mapping between $\mathbb{S}^3$ and $\mathbb{R}^3$. Instead, we exploit a local linearization technique to avoid the singularity problem. The exponential and logarithmic maps provide a natural, non-singular parameterization for "small" angular displacements. Thus, the parameterization of angular displacements is singularity-free if the geodesic distance between each pair of successive points is less than π, that is, $\|\log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})\| < \pi$ for all i. It should be noted that a 3-dimensional vector is an exact representation of the angular displacement, not an approximation.

C. Cumulative vs. Non-cumulative Linearization

Fang *et al.* [5] also employed the idea of the local parameterization to design an orientation filter $\tilde{\mathcal{H}}$ in which a given filter $\mathcal{F}$ is directly applied to a sequence of angular displacement vectors



(a) Motion capture data

(b) $\|\omega'(t)\|^2$ (radian/s^2)

(c) Smoothing by $\mathcal{H}^S$ ($\lambda = 1.0$)

(d) $\|\omega'(t)\|^2$ (radian/s^2)

(e) Blurring by $\mathcal{H}^B$

(f) $\|\omega'(t)\|^2$ (radian/s^2)

(g) Sharpening by $\mathcal{H}^U$ ($\lambda = 1.0$)

(h) $\|\omega'(t)\|^2$ (radian/s^2)

Fig. 5. Experimental results

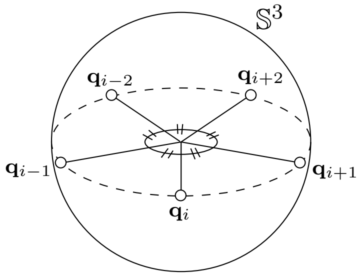


Fig. 6. A singular configuration of re-normalization

as follows:

$$\tilde{\mathcal{H}}(\mathbf{q}_i) = \mathbf{q}_0 \prod_{j=0}^{i-1} \exp\Big(\mathcal{F}(\omega_j)\Big), \tag{28}$$

where $\omega_j = \log(\mathbf{q}_j^{-1}\mathbf{q}_{j+1})$. However, this filter has drawbacks. To illustrate, consider the support of the filter. Since the responses of $\mathcal{F}$ are explicitly accumulated to produce the response of $\tilde{\mathcal{H}}$, the value of $\tilde{\mathcal{H}}(\mathbf{q}_i)$ is influenced by the non-local, asymmetric neighbors $(\mathbf{q}_0, \cdots, \mathbf{q}_i, \cdots, \mathbf{q}_{i+k})$ of $\mathbf{q}_i$. Thus, small deviations in the support can be accumulated to yield larger discrepancies at the end of the signal as shown in Figure 1. In our orientation filter,

$$\begin{aligned}\mathcal{H}(\mathbf{q}_i) &= \mathbf{q}_i \exp\Big(\mathcal{F}(\mathbf{p}_i) - \mathbf{p}_i\Big) \\ &= \mathbf{q}_i \exp\Big(\sum_{m=-k}^{m=k} b_m \omega_{i+m}\Big) \end{aligned} \tag{29}$$

so that we consider only the displacement $(\mathcal{F}(\mathbf{p}_i) - \mathbf{p}_i)$ gained by a given filter. Unlike the direct filter response $\mathcal{F}(\mathbf{p}_i)$, this displacement is computed from a local neighborhood of $\mathbf{q}_i$ without explicitly evaluating $\mathbf{p}_i$ and $\mathcal{F}(\mathbf{p}_i)$. Therefore, the deviation at a frame is not propagated to other frames and thus the filter responses are time-invariant.

D. Multi-linear Methods

As mentioned earlier, multi-linear methods guarantee all properties that we expect orientation filters to have. Multi-linear methods are coordinate-invariant because *slerp* is coordinate-invariant. Time-invariance and symmetry are also guaranteed for multi-linear methods that have a local support and a symmetric mask. However, their major drawback is that they are less efficient than linearization methods. The de Casteljau-like algorithm takes $O(k^2)$ *slerp* for each point, where k is the size of filter mask. The most efficient version of multi-linear methods needs at least $(k-1)$ *slerp* computations for each point. This computation cost is much heavier than that of linearization methods which compute a pair of exponential and logarithmic maps for each point; note that the computation cost for *slerp* is equivalent to the cost for evaluating a pair of exponential and logarithmic maps. Instead, linearization methods require additional storage for keeping the results of transformation. Our algorithm requires $O(n)$ space for storing a sequence of angular displacement vectors, where n is the number of input quaternion points.

VIII. CONCLUSION

Filter masking is a simple, powerful technique for digital signal processing. In this paper, we presented a novel scheme to design orientation filters that correspond to given filter masks. We showed that our orientation filters are computationally efficient and have such desirable properties as coordinate-invariance, time-invariance, and symmetry. We also provided examples that perform smoothing and sharpening on orientation data.

ACKNOWLEDGEMENT

This work was supported by the NRL (National Research Laboratory) program of KISTEP (Korea Institute of Science & Technology Evaluation and Planning). We thank Michael Cohen at Microsoft Research for his valuable comments and Min Gyu Choi at KAIST for his careful reading. We thank Audrey Hodgins and Jessica Hodgins for their writing advice.

REFERENCES

[1] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through HMD. *Computer Graphics (Proceedings of SIGGRAPH 94)*, 28:197–204, July 1994.

[2] A. Barr, B. Currin, S. Gabriel, and J. Hughes. Smooth interpolation of orientations with angular velocity constraints. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26:313–320, July 1992.

[3] A. Bruderlin and L. Williams. Motion signal processing. *Computer Graphics (Proceedings of SIGGRAPH 95)*, 29:97–104, August 1995.

[4] M. Curtis. *Matrix Groups*. Springer-Verlag, 1972.

[5] Y. C. Fang, C. C. Hsieh, M. J. Kim, J. J. Chang, and T. C. Woo. Real time motion fairing with unit quaternions. *Computer-Aided Design*, 30(3):191–198, March 1998.

[6] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, 1963.

[7] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1993.

[8] S. Grassia. Practical parameterization of rotations using the exponential map. *The Journal of Graphics Tools*, 3(3):29–48, 1998.

[9] G. Hanotaux and B. Peroche. Interactive control of interpolations for animation and modeling. In *Proceedings of Graphics Interface '93*, pages 201–208, 1993.

[10] C. C. Hsieh, Y. C. Fang, M. E. Wang, C. K. Wang, M. J. Kim, S. Y. Shin, and T. C. Woo. Noise smoothing for VR equipment in quaternions. *IIE Transactions*, 30:581–587, 1998.

[11] B. Jähne. *Digital Image Processing: Concepts, Algorithms and Scientific Applications*. Springer-Verlag, 1992.

[12] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[13] J. Johnstone and J. Williams. Rational control of orientation for animation. In *Proceedings of Graphics Interface '95*, pages 179–186, 1995.

[14] M. J. Kim. *General Schemes for Unit Quaternion Curve Construction*. PhD thesis, *DCS88343*, Department of Computer Science, KAIST, 1996.

[15] M. J. Kim, C. C. Hsieh, M. E. Wang, C. K. Wang, Y. C. Fang, and T. C. Woo. Noise smoothing for VR equipment in the quaternion space. In *Proceedings of the Symposium on Virtual Reality in Manufacturing Research and Education*, October 1996.

[16] M. J. Kim, M. S. Kim, and S. Y. Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. *Computer Graphics (Proceedings of SIGGRAPH 95)*, 29:369–376, August 1995.

[17] J. Lee and S. Y. Shin. Motion fairing. In *Proceedings of Computer Animation '96*, pages 136–143, June 1996.

[18] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. *Computer Graphics (Proceedings of SIGGRAPH 99)*, pages 39–48, August 1999.

[19] J. Lee and S. Y. Shin. Multiresolution motion analysis with applications. In *Proceedings of the International Workshop on Human Modeling and Animation, Seoul*, pages 131–143, June 2000.

[20] J. Lee and S. Y. Shin. A coordinate-invariant approach to multiresolution motion analysis. *Graphical Models*, to appear, 2001.

[21] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[22] F. C. Park and B. Ravani. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics*, 16(3):277–295, 1997.

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

[24] R. Ramamoorthi and A. H. Barr. Fast construction of accurate quaternion splines. *Computer Graphics (Proceedings of SIGGRAPH 97)*, 31:287–292, August 1997.

[25] K. Shoemake. Animating rotation with quaternion curves. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19:245–254, 1985.

[26] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. *Computer Graphics (Proceedings of SIGGRAPH 95)*, 29:91–96, August 1995.

Jehee Lee received the BS degree in Computer Science from Korea Institute of Technology, Korea, in 1993 and MS and PhD degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1995 and 2000, respectively. He is currently a postdoctoral fellow of the Robotics Institute at Carnegie Mellon University. His primary research interests are computer graphics, motion editing and retargeting, multiresolution analysis and synthesis, and motion signal processing.

Sung Yong Shin received his BS degree in industrial engineering in 1970 from Hanyang University, Seoul, Korea, and his MS and PhD degrees in industrial engineering from the University of Michigan in 1983 and 1986, respectively. He is a professor in the Department of Electrical Engineering and Computer Science at the Korea Advanced Institute of Science and Technology, Taejon, Korea. He has been working at KAIST since 1987. At KAIST, he teaches computer graphics and computational geometry. He also leads a computer graphics research group that has been nominated as a national research laboratory by the Government of Korea. His recent research interests include motion capture and retargeting, performance-based computer animation, real-time rendering, and collision detection.