A Coordinate-Invariant Approach to Multiresolution Motion Analysis

Jehee Lee

Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

and

Sung Yong Shin

Department of Electric Engineering and Computer Science
Korea Advanced Institute of Science and Technology
373-1 Kungsong-dong Yusong-gu, Taejon 305-701, Korea.

Received Febrary 14, 2001; accepted March 23, 2001

Multiresolution motion analysis has gained considerable research interest as a unified framework to facilitate a variety of motion editing tasks. Within this framework, motion data are represented as a collection of coefficients that form a coarse-to-fine hierarchy. The coefficients at the coarsest level describe the global pattern of a motion signal, while those at fine levels provide details at successively finer resolutions. Due to the inherent nonlinearity of the orientation space, the challenge is to generalize multiresolution representations for motion data that contain orientations as well as positions. Our goal is to develop a multiresolution analysis method that guarantees *coordinate-invariance* without singularity. To do so, we employ two novel ideas: hierarchical displacement mapping and motion filtering. Hierarchical displacement mapping provides an elegant formulation to describe positions and orientations in a coherent manner. Motion filtering enables us to separate motion details level-by-level to build a multiresolution representation in a coordinate-invariant way. Our representation facilitates multiresolution motion editing through level-wise coefficient manipulation that uniformly addresses issues raised by motion modification, blending, and stitching.

Keywords: Multiresolution analysis, Coordinate-invariance, Hierarchical techniques, Motion editing, Motion signal processing.

1. INTRODUCTION

Motion capture systems offer a convenient means of acquiring realistic motion data, that is, capturing live motion. Due to the success of those systems, realistic, highly detailed motion data are rapidly becoming popular in computer graphics. Archives of motion clips are also commercially available. Such data sets are used widely in a variety of applications including animation film production, interactive character animation for television, and video games.

Although high quality motion clips are relatively easy to obtain by virtue of motion capture techniques, crafting various animations of arbitrary length with available motion clips remains difficult and requires such specialized tools as interactive editing, blending, stitching, smoothing, enhancement/attenuation, up/down-sampling, and compression. Bruderlin and Williams [1] demonstrated that multiresolution analysis can be a unified framework to implement those tools. The basic idea is to represent motion data (or signals) as a collection of coefficients that form a coarse-to-fine hierarchy. The coefficients at the coarsest level (or resolution) describe the global pattern of a motion signal, while those at fine levels provide details at successively finer resolutions. With the representation, existing motion data can be edited interactively by amplifying/attenuating particular frequency bands and new motions can also be generated by the band-wise blending of existing motions.

Although well-established methods exist for multiresolution analysis in vector spaces, the majority of these methods do not easily generalize in a uniform way for manipulating motion data that contain orientations as well as positions. For example, the vector space methods could be adapted to handle orientation data represented by Euler angles; however, Euler angle parameterization has a singularity that incurs serious artifacts for most signal processing techniques as well as for multiresolution analysis. To avoid such problems, a nonsingular orientation representation, such as rotation matrices or unit quaternions, can be employed. Due to the inherent nonlinearity of the orientation space, however, the challenge is to generalize the results of multiresolution analysis in vector spaces for the orientation space.

The major concern in developing a new multiresolution analysis method is to guarantee such important properties as *coordinate-invariance*. A multiresolution representation is coordinate-invariant if its coefficients are not influenced by the choice of the coordinate system in which the original motion signal is represented. We can also define the coordinate-invariance for such motion editing operations as smoothing, blending, and stitching, to yield consistent results independent of coordinate systems. Coordinate-invariance is of significance not only in theoretical viewpoints but also in practical situations. Suppose, for example, that two identical motion clips are placed at different positions in a reference frame and that we apply the same operation to modify those motions. In this situation, a common expectation is that the identical results will occur independently of the positions of the motion clips. A coordinate-invariant operation guarantees this expectation.

In this paper, we present a new approach to multiresolution motion analysis that is nonsingular and guarantees coordinate-invariance. To do so, we employ two ideas, hierarchical displacement mapping and motion filtering, that provide an elegant formulation to handle positions and orientations in a coherent manner without yielding singularity. Our multiresolution representation consists of a coarse base signal and detail coefficients that form a hierarchy of motion displacement maps. Displacement mapping was originally invented for warping a canned motion while preserving its fine details [1, 27]. In our context, displacement maps are used for adding details level-by-level to the base signal to reproduce the original motion encoded in a multiresolution representation. Our construction algorithm relies on a novel scheme for designing time-domain filters for motion data [16]. With those filters, we are able to construct a multiresolution representation by separating motion details level-by-level in a coordinate-invariant way.

The remainder of the paper is organized as follows. After reviewing the relevant previous work, we describe a hierarchical structure for storing a motion signal and explain how to construct it in Section 3. In Section 4, we provide a proof for the coordinate-invariance of our multiresolution representation. In Section 5, experimental results are demonstrated. Finally, we conclude the paper in Section 6.

2. RELATED WORK

The notion of multiresolution analysis was initiated by Burt and Adelson [2] who introduced a multiresolution image representation, the Gauss-Laplacian pyramid, to facilitate such operations as seamless merging of image mosaics and temporal dissolving between images. Their underlying idea was to decompose an image into a set of band-pass filtered component images, each representing a different band of spatial frequency. This idea was further elaborated by Mallat [18] to establish a multiresolution analysis for continuous functions in connection with wavelet transformation.

Multiresolution techniques have been extensively exercised in computer graphics for curve and surface editing, polygonal mesh editing, image editing and querying, texture analysis and synthesis, video editing and viewing, image and surface compression, global illumination, and variational modeling [23]. These techniques have been used in motion editing and synthesis as well. Liu *et al.* [17] reported that adaptive refinement with hierarchical wavelets provides a significant speed-up for spacetime optimization. Bruderlin and Williams [1] adopted a digital filter-bank technique to address multiresolution analysis of discrete motion data. Their hierarchical representation of a motion with frequency bands allows level-by-level editing of motion characteristics.

LTI (linear time-invariant) filters play a central role in digital filter-bank techniques. Recently, there have been increasing efforts to generalize LTI filters for motion data that contain orientations as well as positions. While a great deal of research results are available for position data, the research for orientation data has recently been emerging. Lee and Shin [14] formulated rotation smoothing as a nonlinear optimization problem and derived smoothing operators from a series of fairness functionals defined on orientation data. Hsieh *et al.* [9, 10] 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. Fang *et al.* [5] applied a low-pass filter to the estimated angular velocity of an input signal to reconstruct a smooth angular motion by integrating the filter responses. More recently, Lee and Shin [16] presented a general scheme for designing an orientation filter which is computationally efficient and guarantees such important filter properties as *coordinate-invariance*, *time-invariance*, and *symmetry*.

One of the most important issues in motion editing is the development of tools that can be used for manipulating highly detailed motion data. Witkin and Popović [27] introduced motion warping (also called displacement mapping) as a means of editing motion data while preserving its fine details. Unuma *et al.* [25] used Fourier analysis techniques to interpolate and extrapolate motion data in the frequency domain. Rose *et al.* [20] suggested a semiautomatic scheme for stitching motion clips seamlessly. Wiley and Hahn [26] and Guo and Robergé [8] investigated spatial domain approaches to interpolate linearly a set of example motions. Rose *et al.* [19] adopted a multidimensional interpolation method to blend multiple motions together. Gleicher [6, 7] adapted the spacetime optimization formulation for editing motion with a set of kinematic spacetime constraints. Lee and

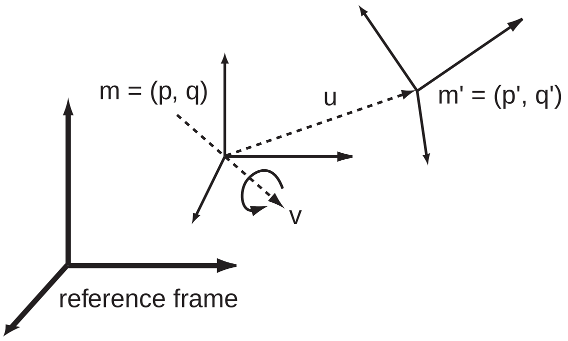

FIG. 1. Motion displacements

Shin [15] introduced hierarchical displacement mapping for adaptively refining a motion to meet spacetime constraints.

3. MULTIRESOLUTION REPRESENTATION

In this section, we present a multiresolution representation of motion. It consists of a coarse base signal and detail coefficients that form a hierarchy of motion displacement maps. The displacement map at each level includes a sequence of coefficients. The coefficients at the base level determine the overall shape of the motion signal, and its details are added successively with those at fine levels. In Section 3.1, we explain displacement mapping and its mathematical backgrounds. In section 3.2, motion filtering is briefly described. In section 3.3, we elaborate a general framework of multiresolution analysis based on hierarchical displacement mapping and motion filtering.

3.1. Displacement Mapping

The pose of an articulated figure can be specified by its joint configurations in addition to the position and orientation of the root segment. For uniformity, we assume that the configuration of each joint is given by a 3-dimensional rigid transformation. Then, we can describe the degrees of freedom at every body segment as a pair of a vector in $\mathbb{R}^3$ and a unit quaternion in $\mathbb{S}^3$. The *motion data* for an articulated figure comprise a bundle of *motion signals*. Every signal consists of a sequence of frames, $\{(\mathbf{p}_i, \mathbf{q}_i) \in \mathbb{R}^3 \times \mathbb{S}^3\}$, each of which corresponds to the position and orientation of a body segment. A frame $(\mathbf{p}_i, \mathbf{q}_i)$ specifies a rigid transformation $T_{(\mathbf{p}_i,\mathbf{q}_i)}$ that maps a point in $\mathbb{R}^3$ to another in $\mathbb{R}^3$:

$$T_{(\mathbf{p}_i,\mathbf{q}_i)}(\mathbf{x}) = \mathbf{q}_i \mathbf{x} \mathbf{q}_i^{-1} + \mathbf{p}_i. \tag{1}$$

Here, $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ is considered a purely imaginary quaternion $(0, x, y, z) \in \mathbb{R}^4$.

Given two motion signals $\mathbf{m} = \{(\mathbf{p}_i, \mathbf{q}_i) \in \mathbb{R}^3 \times \mathbb{S}^3\}$ and $\mathbf{m}' = \{(\mathbf{p}'_i, \mathbf{q}'_i) \in \mathbb{R}^3 \times \mathbb{S}^3\}$, we define their motion displacement $\mathbf{d} = \{(\mathbf{u}_i, \mathbf{v}_i) \in \mathbb{R}^3 \times \mathbb{R}^3\}$ measured in a local (body-fixed) coordinate system such that $T_{(\mathbf{p}'_i,\mathbf{q}'_i)} = T_{(\mathbf{p}_i,\mathbf{q}_i)} \circ T_{(\mathbf{u}_i,\exp(\mathbf{v}_i))}$. In a geometric viewpoint, the motion frame $\mathbf{m}_i = (\mathbf{p}_i, \mathbf{q}_i)$ at a specific time instance is transformed to a new frame $\mathbf{m}'_i = (\mathbf{p}'_i, \mathbf{q}'_i)$ through the rotation of $\mathbf{m}_i$ about the axis of $\mathbf{v}_i$ by the two times of its magnitude followed by the translation along $\mathbf{u}_i$ (Figure 1). In later discussion, we introduce two operators $\oplus$ and $\ominus$ such that $\mathbf{m}' = \mathbf{m} \oplus \mathbf{d}$ and $\mathbf{d} = \mathbf{m}' \ominus \mathbf{m}$. From

Equation (1), we have

$$\begin{aligned}(\mathbf{p}'_i, \mathbf{q}'_i) &= (\mathbf{p}_i, \mathbf{q}_i) \oplus (\mathbf{u}_i, \mathbf{v}_i) \\ &= (\mathbf{q}_i \mathbf{u}_i \mathbf{q}_i^{-1} + \mathbf{p}_i, \mathbf{q}_i \exp(\mathbf{v}_i))\end{aligned} \tag{2}$$

or conversely

$$\begin{aligned}(\mathbf{u}_i, \mathbf{v}_i) &= (\mathbf{p}'_i, \mathbf{q}'_i) \ominus (\mathbf{p}_i, \mathbf{q}_i) \\ &= (\mathbf{q}_i^{-1}(\mathbf{p}'_i - \mathbf{p}_i)\mathbf{q}_i, \log(\mathbf{q}_i^{-1}\mathbf{q}'_i)),\end{aligned} \tag{3}$$

where $\exp(\mathbf{v}_i)$ denotes a 3-dimensional rotation about the axis $\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \in \mathbb{R}^3$ by angle $2\|\mathbf{v}_i\| \in \mathbb{R}$. The definition of a motion displacement map in the above equation has two advantages. First, because both linear and angular displacement vectors are represented in the body-fixed coordinate frame, the motion displacement map is not influenced by the choice of a global reference frame in which motion signals are represented. Second, we do not need to distinguish position and orientation data in the displacement map because both have an identical form: Note that a motion frame consists a heterogeneous pair, a 3-dimensional vector and a unit quaternion, while a motion displacement consists a homogeneous pair of 3-dimensional vectors.

3.2. Motion Filtering

Given a vector-valued signal $\mathbf{p}_i \in \mathbb{R}^3$ and a filter mask $(a_{-k}, \cdots, a_0, \cdots, a_k)$, the basic idea of 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

$$\mathcal{F}(\mathbf{p}_i) = a_{-k}\mathbf{p}_{i-k} + \cdots + a_0\mathbf{p}_i + \cdots + a_k\mathbf{p}_{i+k}. \tag{4}$$

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*, or *symmetry*.

Lee and Shin [16] presented a general scheme of constructing a time-domain filter for orientation data that is a quaternion counterpart of Equation (4). Given a filter mask $(a_{-k}, \cdots, a_0, \cdots, a_k)$ of which coefficients are summed up to one, an orientation filter can be defined as

$$\mathcal{H}(\mathbf{q}_i) = \mathbf{q}_i \exp\left(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{q}_{i+m+1})\right), \tag{5}$$

where

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

Clearly, the unitariness of filter responses is guaranteed, because the unit quaternion space is closed under the quaternion multiplication. Furthermore, the filter $\mathcal{H}$ for orientation data inherits important properties from its vector counterpart given in Equation (4). Here, we summarize the properties of $\mathcal{H}$ without proofs. Detailed discussion is found in Lee and Shin [16]. First, $\mathcal{H}$ is invariant under both local and global coordinate transformations, that

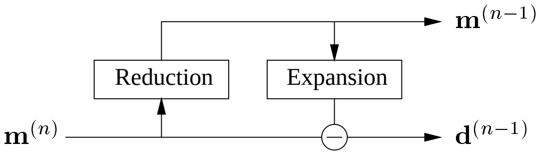

FIG. 2. Wiring diagram of the multiresolution analysis

is, $\mathbf{a}\mathcal{H}(\mathbf{q}_i)\mathbf{b} = \mathcal{H}(\mathbf{a}\mathbf{q}_i\mathbf{b})$ for any $\mathbf{a}$ and $\mathbf{b} \in \mathbb{S}^3$. Due to this property, $\mathcal{H}$ yields identical results independent of the coordinate system in which the orientation data are represented. Second, $\mathcal{H}$ is time-invariant, that is, its filter response does not depend on the position in the signal. Finally, $\mathcal{H}$ is symmetric, if its mask coefficients are symmetric.

Inherent ambiguity exists 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 the signs of quaternion points must be corrected consistently before filtering. We determine the sign of each point in the signal such that the point is placed near its adjacent neighbors. To do so, we initially 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

$$\omega_i = \begin{cases} \omega_{-i}, & \text{if } i < 0, \\ \omega_{2n-i-2}, & \text{if } i \ge n. \end{cases} \tag{6}$$

3.3. Construction

Our multiresolution representation for a motion signal $\mathbf{m} = \mathbf{m}^{(N)}$ is defined by a series of successively refined signals $\mathbf{m}^{(0)}, \mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(N-1)}$ together with a series of displacement maps $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(N-1)}$. The construction of the multiresolution representation is based on two basic operations: reduction and expansion (Figure 2). The expansion $\mathcal{E}$ is achieved by a subdivision operation that can be considered as up-sampling followed by smoothing. The reduction $\mathcal{R}$ is a reverse operation, that is, smoothing followed by down-sampling. Smoothing operations avoid aliasing caused by down-sampling and interpolating missing information for up-sampling.

Our construction algorithm begins with the original motion $\mathbf{m}^{(N)}$ to compute its simplified versions and their corresponding displacement maps successively in fine-to-coarse order. Suppose that we are now at the n-th level for $0 \le n \le N-1$. Given a signal $\mathbf{m}^{(n+1)}$, we can compute a coarser signal $\mathbf{m}^{(n)}$ by reduction. The expansion of $\mathbf{m}^{(n)}$ interpolates the missing information to approximate the original signal $\mathbf{m}^{(n+1)}$. Thus, the

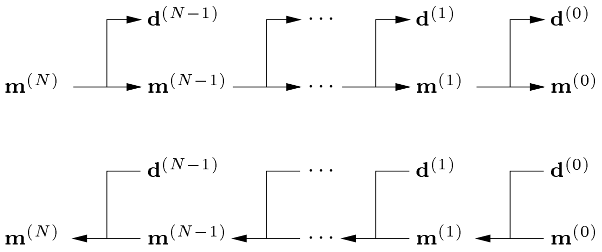

FIG. 3. Decomposition (upper) and reconstruction (lower)

difference between them is expressed as a displacement map $\mathbf{d}^{(n)}$:

$$\mathbf{m}^{(n)} = \mathcal{R}\mathbf{m}^{(n+1)}, \tag{7}$$

$$\mathbf{d}^{(n)} = \mathbf{m}^{(n+1)} \ominus \mathcal{E}\mathbf{m}^{(n)}. \tag{8}$$

Cascading these operations until there remains a sufficiently small number of frames in the motion signal, we can construct a multiresolution representation that includes the coarse base signal $\mathbf{m}^{(0)}$ and a series of displacement maps as shown in Figure 3(upper). Conversely, the original signal $\mathbf{m}^{(N)}$ can be reconstructed from the multiresolution representation by recursively adding the displacement map at each level to the expansion of the signal at the same level, that is,

$$\begin{aligned}\mathbf{m}^{(N)} &= \mathcal{E}\mathbf{m}^{(N-1)} \oplus \mathbf{d}^{(N-1)} \\ &= \mathcal{E}(\mathcal{E}\mathbf{m}^{(N-2)} \oplus \mathbf{d}^{(N-2)}) \oplus \mathbf{d}^{(N-1)} \\ &= \mathcal{E}(\mathcal{E}\cdots(\mathcal{E}\mathbf{m}^{(0)} \oplus \mathbf{d}^{(0)})\cdots \oplus \mathbf{d}^{(N-2)}) \oplus \mathbf{d}^{(N-1)},\end{aligned} \tag{9}$$

as shown in Figure 3(lower).

Several alternative choices can be used to implement the reduction and expansion operations. In the original work of Gauss-Laplacian image pyramids [2], Gaussian filters (which are approximated by binomial filter masks) are used for both operations to avoid aliasing effects that are mainly incurred by discontinuity in the input image. Unlike digital images, motion data have C^0 continuity and thus we adopt Laplacian smoothing and interpolatory subdivision for designing reduction and expansion operations, respectively, which preserve the original signal better than Gaussian filtering

Reduction. Given a detailed signal $\mathbf{m}^{(n+1)}$, the reduction operator $\mathcal{R}$ generates its simplified version $\mathbf{m}^{(n)}$ at a coarser resolution by applying a smoothing filter to $\mathbf{m}^{(n+1)}$ and then removing every other frame to down-sample the signal. Hence, $\mathcal{R}$ can be regarded as the composition of a down-sampling operator $\mathcal{D}$ of factor two and a smoothing operator $\mathcal{H}^R$, that is,

$$\mathbf{m}^{(n)} = \mathcal{R}\mathbf{m}^{(n+1)} = (\mathcal{D} \circ \mathcal{H}^R)\mathbf{m}^{(n+1)}. \tag{10}$$

A popular way to implement a smoothing operator $\mathcal{H}^R$ is to adopt a diffusion process that leads to a local update rule

$$\mathbf{p}_i \leftarrow \mathbf{p}_i - \lambda L^j \mathbf{p}_i, \tag{11}$$

where λ is a diffusion coefficient and L is a Laplacian operator [12, 24]. Filtering with this rule disperses small perturbations rapidly while the original shape is degraded only slightly. Here, Laplacian operators can be estimated for discrete signals by replacing differential operators with forward divided difference operators such that $L^j = \Delta^{2j}$, where

$$\begin{aligned} \Delta^1 \mathbf{p}_i &= \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{t_{i+1} - t_i}, \\ \Delta^j \mathbf{p}_i &= \frac{\Delta^{j-1}\mathbf{p}_{i+1} - \Delta^{j-1}\mathbf{p}_i}{t_{i+j} - t_i}, \quad \text{for } j > 1. \end{aligned} \tag{12}$$

This update rule yields an affine-invariant filter mask that can be generalized for orientation data by using Equation (5). For example, by adopting the second Laplacian operator L^2 and letting $\lambda = 1$, we have a filter mask $\frac{1}{24}(-1, 4, 18, 4, -1)$ and its corresponding filter,

$$(\mathbf{p}'_i, \mathbf{q}'_i) = \mathcal{H}^R(\mathbf{p}_i, \mathbf{q}_i). \tag{13}$$

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

$$\begin{aligned} \mathbf{p}'_i &= \frac{1}{24}\left(-\mathbf{p}_{i-2} + 4\mathbf{p}_{i-1} + 18\mathbf{p}_i + 4\mathbf{p}_{i+1} - \mathbf{p}_{i+2}\right), \\ \mathbf{q}'_i &= \mathbf{q}_i \exp\left(\frac{1}{24}(\omega_{i-2} - 3\omega_{i-1} + 3\omega_i - \omega_{i+1})\right). \end{aligned}$$

Expansion. Given a coarse signal $\mathbf{m}^{(n)}$, the expansion operator $\mathcal{E}$ approximates a corresponding signal $\mathbf{m}^{(n+1)}$ at a higher resolution by interpolation followed by error compensation:

$$\mathbf{m}^{(n+1)} = \mathcal{E}\mathbf{m}^{(n)} \oplus \mathbf{d}^{(n)} = (\mathcal{H}^E \circ \mathcal{U})\mathbf{m}^{(n)} \oplus \mathbf{d}^{(n)}, \tag{14}$$

where $\mathbf{d}^{(n)}$ represents an approximation error. To obtain a smoother signal of higher resolution, a cubic polynomial is a good choice for trading off smoothness for efficiency. Thus, the operator $\mathcal{E}$ can be achieved by a *four-point interpolatory subdivision* scheme that maps a sequence of motion frames $\mathbf{m}^{(n)} = \{(\mathbf{p}_i^n, \mathbf{q}_i^n)\}$ to a refined sequence $\mathbf{m}^{(n+1)} = \{(\mathbf{p}_i^{n+1}, \mathbf{q}_i^{n+1})\}$, where the even numbered frames $(\mathbf{p}_{2i}^{n+1}, \mathbf{q}_{2i}^{n+1})$ at level $n+1$ are the frames $(\mathbf{p}_i^n, \mathbf{q}_i^n)$ at level n, and the odd numbered frames $(\mathbf{p}_{2i+1}^{n+1}, \mathbf{q}_{2i+1}^{n+1})$ are newly inserted between old frames.

To generalize the subdivision scheme to the orientation data, the scheme should be considered in two separate phases, that is, up-sampling $\mathcal{U}$ followed by smoothing $\mathcal{H}^E$ (Figure 4). At the up-sampling phase, the odd numbered frames $(\mathbf{p}_{2i+1}^{n+1}, \mathbf{q}_{2i+1}^{n+1})$ are inserted halfway between two successive old frames using (spherical) linear interpolation. Assuming that the motion frames are sampled uniformly, we have $\mathbf{p}_{2i+1}^{n+1} = \frac{1}{2}\mathbf{p}_i^n + \frac{1}{2}\mathbf{p}_{i+1}^n$ and $\mathbf{q}_{2i+1}^{n+1} = \text{slerp}_{\frac{1}{2}}(\mathbf{q}_i^n, \mathbf{q}_{i+1}^n)$. Here, $\text{slerp}_t(\mathbf{q}_1, \mathbf{q}_2)$ denotes a spherical linear interpolation between two unit quaternion points $\mathbf{q}_1$ and $\mathbf{q}_2$ with interpolation parameter t, that is, $\text{slerp}_t(\mathbf{q}_1, \mathbf{q}_2) = \mathbf{q}_1 \exp(t \cdot \log(\mathbf{q}_1^{-1}\mathbf{q}_2))$ [22]. At the smoothing phase, the smoothing operator is applied only to the newly inserted points with a subdivision mask

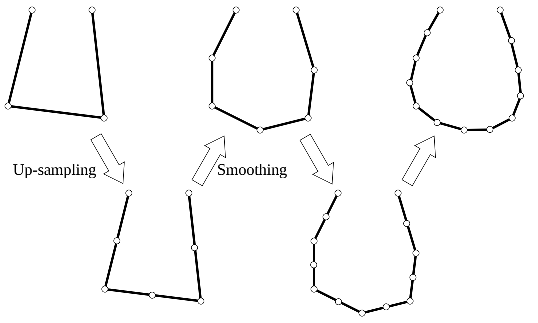

FIG. 4. Interpolatory subdivision

$(-\frac{1}{16}, 0, \frac{9}{16}, 0, \frac{9}{16}, 0, -\frac{1}{16})$ to yield the refined data as follows:

$$\begin{aligned} \mathbf{p}_{2i}^{n+1} &= \mathbf{p}_i^n, \\ \mathbf{p}_{2i+1}^{n+1} &= \frac{1}{16}(-\mathbf{p}_{2i-2}^{n+1} + 9\mathbf{p}_{2i}^{n+1} + 9\mathbf{p}_{2i+2}^{n+1} - \mathbf{p}_{2i+4}^{n+1}) \\ &= \frac{1}{16}(-\mathbf{p}_{i-1}^n + 9\mathbf{p}_i^n + 9\mathbf{p}_{i+1}^n - \mathbf{p}_{i+2}^n). \end{aligned} \tag{15}$$

The point $\mathbf{p}_{2i+1}^{n+1}$ thus obtained locates halfway between $\mathbf{p}_i^n$ and $\mathbf{p}_{i+1}^n$ on the cubic polynomial curve interpolating four neighboring points $\mathbf{p}_{i-1}^n$, $\mathbf{p}_i^n$, $\mathbf{p}_{i+1}^n$ and $\mathbf{p}_{i+2}^n$ [3, 4]. Similarly, we can use the smoothing operator in Equation (5) with the same subdivision mask to obtain the orientation version of the subdivision scheme as follows:

$$\begin{aligned} \mathbf{q}_{2i}^{n+1} &= \mathbf{q}_i^n, \\ \mathbf{q}_{2i+1}^{n+1} &= \text{slerp}_{\frac{1}{2}}(\mathbf{q}_i^n, \mathbf{q}_{i+1}^n)\exp(\frac{\omega_{i-1}^n - \omega_{i+1}^n}{16}), \end{aligned} \tag{16}$$

where $\omega_i^n = \log((\mathbf{q}_i^n)^{-1}\mathbf{q}_{i+1}^n)$.

If the smoothing filters for reduction and expansion are not induced from a bi-orthogonal wavelet basis, then this construction scheme gives over-representations, as Gauss-Laplacian image pyramids do, in the sense that the decomposition of $\mathbf{m}^{(n)}$ into a coarser signal $\mathbf{m}^{(n-1)}$ and its detail coefficients in $\mathbf{d}^{(n-1)}$ yields extra data to store. For such memory-critical applications as compression and progressive transmission, we can circumvent such extra data by skipping the smoothing step of the reduction operation in a spirit of lazy wavelets [18, 21]. Then, $\mathbf{m}^{(n-1)}$ contains the even frames of $\mathbf{m}^{(n)}$ and thus we have non-zero detail coefficients in $\mathbf{d}^{(n-1)}$ only for odd frames to achieve an exact representation of the same size.

3.4. Extension

Though most motion captured data are sampled at a sequence of time instances of uniform interval, we often need to process non-uniform data to support such tasks as time warping, which aligns motion clips with respect to time [1]. To construct a multiresolution representation for nonuniformly sampled motion data, we further generalize the reduction and expansion operators for a nonuniform setting. For reduction, we can easily derive smoothing masks by estimating discrete Laplacian operators for a nonuniform setting, since the divided difference operator is well defined. For expansion, the coefficients of the subdivision mask are derived from the cubic Lagrange polynomials [13]. The cubic polynomial that interpolates four points $(\mathbf{p}_{i-1}^n, \mathbf{p}_i^n, \mathbf{p}_{i+1}^n, \mathbf{p}_{i+2}^n)$ defined over the knot sequence $[t_{i-1}^m, t_i^n, t_{i+1}^n, t_{i+2}^n]$ can be written as follows:

$$\mathbf{p}(t) = l_{1000}(t)\mathbf{p}_{i-1}^n + l_{0100}(t)\mathbf{p}_i^n + l_{0010}(t)\mathbf{p}_{i+1}^n + l_{0001}(t)\mathbf{p}_{i+2}^n, \tag{17}$$

where the cardinal function $l_{u_0 u_1 u_2 u_3}(t)$ is the unique cubic polynomial that interpolates u_j at t_{i+j-1}^n for $0 \leq j \leq 3$ [11]. Note that Equation (17) is a simple generalization of Equation (15). Therefore, we can obtain a subdivision mask $(l_{1000}(t_{2i+1}^{n+1}), 0, l_{0100}(t_{2i+1}^{n+1}), 0, l_{0010}(t_{2i+1}^{n+1}), 0, l_{0001}(t_{2i+1}^{n+1}))$ to compute $\mathbf{p}_{2i+1}^{n+1}$ and $\mathbf{q}_{2i+1}^{n+1}$.

Proper boundary handling is required for the subdivision scheme in either a uniform or nonuniform setting. At the left boundary, for example, we determine $\mathbf{p}_1^{n+1}$ from the cubic polynomial that interpolates the four left-most points $\mathbf{p}_0^n$, $\mathbf{p}_1^n$, $\mathbf{p}_2^n$ and $\mathbf{p}_3^n$ of the original sequence $\mathbf{m}^{(n)}$. For orientation components, $\mathbf{q}_1^{n+1}$ can also be computed with the filter mask induced from the interpolating polynomial.

4. COORDINATE INVARIANCE

In this section, we will show that our multiresolution representation is coordinate-invariant. Coordinate-invariance can be explained in several ways. The first is that identical motion clips placed at different positions in a reference frame give identical displacement maps in their multiresolution representations. Coordinate-dependent information is retained only in the base signal. To put it another way, we consider a coordinate transformation that consists of 3-dimensional rotation followed by translation. If we apply that coordinate transformation to the base signal of a multiresolution representation and reconstruct the motion signal from the transformed representation, then that signal will be identical to the one obtained by applying the transformation to the original signal.

To prove the coordinate-invariance of our multiresolution representation, we need to verify the invariance involved in motion displacement mapping and filtering. First, because motion displacements are measured in a body-fixed coordinate frame, displacement maps are independent of the choice of the global reference frame. Let $\mathcal{T}_{(\mathbf{t},\mathbf{r})} : \mathbb{R}^3 \times \mathbb{S}^3 \to \mathbb{R}^3 \times \mathbb{S}^3$ be a coordinate transformation such that

$$\mathcal{T}_{(\mathbf{t},\mathbf{r})}\mathbf{m} = (\mathbf{r}\mathbf{p}\mathbf{r}^{-1} + \mathbf{t}, \mathbf{r}\mathbf{q}) \tag{18}$$

for a motion signal $\mathbf{m} = (\mathbf{p}, \mathbf{q})$, where $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{r} \in \mathbb{S}^3$. The righthand side of this equation specifies a composite transform, $T_{(\mathbf{t},\mathbf{r})} \circ T_{(\mathbf{p},\mathbf{q})}$. $\mathcal{T}_{(\mathbf{t},\mathbf{r})}$ yields a coordinate transformation relative to the global reference frame. For notational simplicity, we will use $\mathcal{T}$ instead of $\mathcal{T}_{(\mathbf{t},\mathbf{r})}$ when this use does not cause confusion. The following lemma proves the coordinate-invariance of motion displacement mapping.

LEMMA 4.1. *The displacement map* $\mathbf{d}$ *between any two motions* $\mathbf{m}$ *and* $\mathbf{m}'$ *is invariant under global coordinate transformation, that is,* $\mathbf{d} = \mathbf{m}' \ominus \mathbf{m} = \mathcal{T}\mathbf{m}' \ominus \mathcal{T}\mathbf{m}$ *for any coordinate transformation* $\mathcal{T}$.

Proof: From Equations (3) and (18),

$$\begin{aligned}\mathcal{T}_{(\mathbf{t},\mathbf{r})}\mathbf{m}' \ominus \mathcal{T}_{(\mathbf{t},\mathbf{r})}\mathbf{m} &= (\mathbf{r}\mathbf{p}'\mathbf{r}^{-1} + \mathbf{t}, \mathbf{r}\mathbf{q}') \ominus (\mathbf{r}\mathbf{p}\mathbf{r}^{-1} + \mathbf{t}, \mathbf{r}\mathbf{q}) \\ &= \Big(\mathbf{q}^{-1}\mathbf{r}^{-1}(\mathbf{r}\mathbf{p}'\mathbf{r}^{-1} + \mathbf{t} - \mathbf{r}\mathbf{p}\mathbf{r}^{-1} - \mathbf{t})\mathbf{r}\mathbf{q}, \log(\mathbf{q}^{-1}\mathbf{r}^{-1}\mathbf{r}\mathbf{q}')\Big) \\ &= \Big(\mathbf{q}^{-1}(\mathbf{p}' - \mathbf{p})\mathbf{q}, \log(\mathbf{q}^{-1}\mathbf{q}'))\Big) \\ &= \mathbf{m}' \ominus \mathbf{m}.\end{aligned}$$

■

Next, we need to verify the invariance of motion filtering. Due to the favorable properties of our orientation filtering scheme, both reduction and expansion operations are invariant under coordinate transformation as shown in the following lemma.

LEMMA 4.2. *The reduction and expansion operations commute with coordinate transformation, that is, $\mathcal{R}\mathcal{T} = \mathcal{T}\mathcal{R}$ and $\mathcal{E}\mathcal{T} = \mathcal{T}\mathcal{E}$ for any coordinate transformation $\mathcal{T}$.*

Proof: Let $\mathcal{M}$ be a motion filter consisting of the position and orientation filters defined in Equations (4) and (5), respectively. As shown in Equations (10) and (14), reduction and expansion operations are combinations of motion filtering and resampling. Since resampling does not affect the coordinate-invariance, the proof will be complete if we show that $\mathcal{M}$ commutes with $\mathcal{T}$. Since $\sum_{m=-k}^{k} a_m = 1$,

$$\begin{aligned}&\mathcal{T}_{(\mathbf{t},\mathbf{r})}\circ\mathcal{M}(\mathbf{p}_i, \mathbf{q}_i) \\ &\quad= \mathcal{T}_{(\mathbf{t},\mathbf{r})}(\mathcal{F}(\mathbf{p}_i), \mathcal{H}(\mathbf{q}_i)) \\ &\quad= \mathcal{T}_{(\mathbf{t},\mathbf{r})}\Big(\sum_{m=-k}^{k} a_m\mathbf{p}_{i+m}, \mathbf{q}_i \exp(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{q}_{i+m+1}))\Big) \\ &\quad= \Big(\mathbf{r}(\sum_{m=-k}^{k} a_m\mathbf{p}_{i+m})\mathbf{r}^{-1} + \mathbf{t}, \mathbf{r}\mathbf{q}_i \exp(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{q}_{i+m+1}))\Big) \\ &\quad= \Big(\sum_{m=-k}^{k} a_m(\mathbf{r}\mathbf{p}_{i+m}\mathbf{r}^{-1} + \mathbf{t}), \mathbf{r}\mathbf{q}_i \exp(\sum_{m=-k}^{k-1} b_m \log(\mathbf{q}_{i+m}^{-1}\mathbf{r}^{-1}\mathbf{r}\mathbf{q}_{i+m+1}))\Big) \\ &\quad= \mathcal{M}\circ\mathcal{T}_{(\mathbf{t},\mathbf{r})}(\mathbf{p}_i, \mathbf{q}_i).\end{aligned}$$

■

Let $\mathbf{m} = \mathbf{m}^{(N)}$ be the motion signal at the finest level N. Recursively applying Equations (7) and (8), the displacement map at level $0 \le n < N$ is given:

$$\mathbf{d}^{(n)} = \mathcal{R}^{(N-n-1)}\mathbf{m}^{(N)} \ominus \mathcal{E}\mathcal{R}^{(N-n)}\mathbf{m}^{(N)}. \tag{19}$$

The following theorem establishes that the displacement maps are independent of the choice of the reference frame in which the original signal $\mathbf{m}^{(N)}$ is represented.

THEOREM 4.1. *The displacement maps in a multiresolution representation are invariant under global coordinate transformation, that is,*

$$\begin{aligned}\mathbf{d}^{(n)} &= \mathcal{R}^{(N-n-1)}\mathbf{m}^{(N)} \ominus \mathcal{E}\mathcal{R}^{(N-n)}\mathbf{m}^{(N)} \\ &= \mathcal{R}^{(N-n-1)}\mathcal{T}\mathbf{m}^{(N)} \ominus \mathcal{E}\mathcal{R}^{(N-n)}\mathcal{T}\mathbf{m}^{(N)}\end{aligned}$$

for $0 \leq n < N$.

Proof: Applying Lemma 1 and Lemma 2, we have

$$\begin{aligned}\mathbf{d}^{(n)} &= \mathcal{R}^{(N-n-1)}\mathbf{m}^{(N)} \ominus \mathcal{E}\mathcal{R}^{(N-n)}\mathbf{m}^{(N)} \\ &= \mathcal{T}\mathcal{R}^{(N-n-1)}\mathbf{m}^{(N)} \ominus \mathcal{T}\mathcal{E}\mathcal{R}^{(N-n)}\mathbf{m}^{(N)} \\ &= \mathcal{R}^{(N-n-1)}\mathcal{T}\mathbf{m}^{(N)} \ominus \mathcal{E}\mathcal{R}^{(N-n)}\mathcal{T}\mathbf{m}^{(N)}.\end{aligned}$$

■

As an immediate consequence, we can easily prove the following corollary.

COROLLARY 4.1. *Let* $M(\mathbf{m}) = (\mathbf{m}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(N-1)})$ *be the multiresolution representation of* $\mathbf{m}$. *Then,* $M(\mathcal{T}\mathbf{m}) = (\mathcal{T}\mathbf{m}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(N-1)})$.

Proof: Theorem 1 shows that $M(\mathbf{m})$ and $M(\mathcal{T}\mathbf{m})$ have the same sequence of displacement maps. Therefore, the proof will be complete if we show that the base signal of $M(\mathcal{T}\mathbf{m})$ is $\mathcal{T}\mathbf{m}^{(0)}$. Recursively applying Equation (7), we have

$$\mathcal{R}^{(N)}\mathcal{T}\mathbf{m} = \mathcal{T}\mathcal{R}^{(N)}\mathbf{m} = \mathcal{T}\mathbf{m}^{(0)}.$$

■

5. EXPERIMENTS

We have implemented a prototype motion editing system in C++ on top of Windows NT^{TM} and Open Inventor^{TM}. Our model of a human has 43 DOFs that consist of six DOFs for the pelvis position and orientation, and three DOFs for each of the other joints except for the elbows and knees which have a single DOF. Experiments were conducted on a Pentium PC (single Pentium III processor, 500 MHz) with various motion data (both 30 Hz and 24 Hz) that were captured at a commercial studio. In Section 5.1, we apply our multiresolution analysis techniques to implement several motion editing operations such as enhancement/attenuation, blending, and stitching. In Section 5.2, experimental results will be demonstrated to compare our approach to a conventional approach that is not coordinate-invariant.

5.1. Motion Editing Operations

Our multiresolution representation allows us to modify its fine details at each level independently of those at other levels through the level-wise manipulation of detail coefficients. Detail coefficients at every level are represented by a pair of 3-dimensional vectors that correspond to the displacements for position and orientation, respectively. Therefore, the orientation components can be scaled and blended without concerning the unitariness constraint.

Modification. A natural application of multiresolution analysis is the construction of an LOD (level-of-detail) representation of a motion that consists of its several versions at

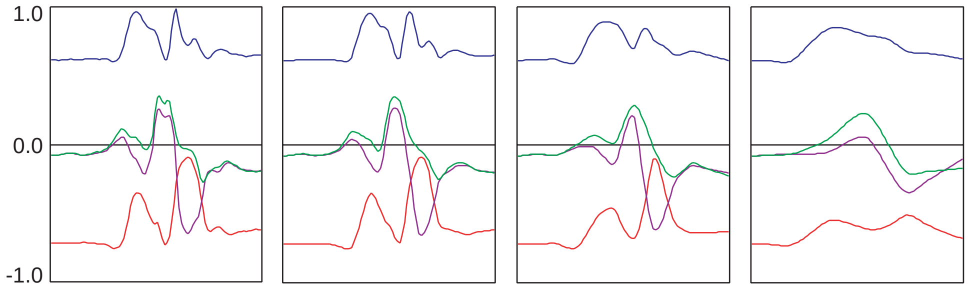

FIG. 5. Level-of-detail generation for a live-captured signal. The four curves represent the change of w-, x-, y-, and z-components, respectively, of a unit quaternion with respect to time. From left to right, original signal and its approximations at successively coarser resolutions

various levels of detail (Figure 5). Given a detailed signal, we can construct a series of successively smoother versions by removing the detail coefficients level by level, starting from the finest level. For continuous transition between levels, we also consider the fractional levels $n + \alpha$ of a motion signal with a blending parameter $0 < \alpha < 1$, that defines a linear interpolation between levels n and $n + 1$. To obtain a fractional level motion, we scale the coefficients at level n by a factor of α, and set all coefficients zero at higher levels. Another promising application is enhancing/attenuating the detailed features of a motion signal to convey various moods or emotions. This application can be achieved through the level-wise scaling of detail coefficients with different scaling factors. For the motion “Jump and Kick” in Figure 6, we multiply the detail coefficients by constant factors to produce the attenuated (left) and enhanced (right) versions, respectively. The enhancement results in a higher jump and kick, while the attenuation conveys a milder emotional mood and softer action. The effects are clearly observed along the trajectories of the feet. Figure 7 shows a motion in which the face is hit by an object. The enhanced and attenuated versions successfully simulate the effects of hard and soft hitting, respectively.

Blending. Motion blending is a popular operation to produce a wide span of motions from a small number of example motions. However, the fine details of motions may be lost, if we blend them grossly without the separation of their features at different scales. Our representation scheme is also useful for blending motion clips together. A particular example in Figure 8 blends three motions of the same size, that is, straight walking $\mathbf{m}_{ws}$, turning while walking $\mathbf{m}_{wt}$, and straight walking with a limp $\mathbf{m}_{ls}$. From these motions, we produce a new motion $\mathbf{m}_{lt}$ that describes turning with a limp. The basic observation is that the global shape of the target motion is similar to $\mathbf{m}_{wt}$ and its fine details are similar to $\mathbf{m}_{ls}$. Therefore, we obtain the base signal $\mathbf{m}_{lt}^{(0)}$ by applying the displacement map $\Phi_{st} = \mathbf{m}_{wt}^{(0)} \ominus \mathbf{m}_{ws}^{(0)}$ to $\mathbf{m}_{ls}^{(0)}$. Similarly, the detail coefficients in $\mathbf{d}_{lt}^{(n)}$ are computed by applying the displacement map $\Phi_{wl}^{(n)} = \mathbf{d}_{ls}^{(n)} \ominus \mathbf{d}_{ws}^{(n)}$ to $\mathbf{d}_{wt}^{(n)}$.

$$\begin{aligned} \mathbf{m}_{lt}^{(0)} &= \mathbf{m}_{ls}^{(0)} \oplus (\mathbf{m}_{wt}^{(0)} \ominus \mathbf{m}_{ws}^{(0)}) \\ \mathbf{d}_{lt}^{(n)} &= \mathbf{d}_{wt}^{(n)} + (\mathbf{d}_{ls}^{(n)} - \mathbf{d}_{ws}^{(n)}), \quad \text{for } 0 < n < N. \end{aligned} \tag{20}$$

Here, Φ_{st} describes how a straight movement is transformed to a turning motion, and $\Phi_{wl}^{(n)}$ describes how normal walking is transformed to limping.

Stitching. In animation production, we are often required to combine canned motion clips together into an animation of unlimited length. A simple approach would be to estimate the linear and angular velocities at the boundaries of each pair of consecutive motion clips and then perform C^1 interpolation. However, obtaining a robust estimate of velocity from live captured signals is difficult because these signals usually oscillate to include fine details that may distinguish the motion of a live creature from the unnatural motion of a robot. Our multiresolution representation can be used for connecting highly detailed signals robustly. Our algorithm is as follows: Given two motion signals $\mathbf{m}_A$ and $\mathbf{m}_B$, their multiresolution representations are constructed to give $\mathbf{M}(A) = (\mathbf{m}_A^{(0)}, \mathbf{d}_A^{(0)}, \cdots, \mathbf{d}_A^{(N-1)})$ and $\mathbf{M}(B) = (\mathbf{m}_B^{(0)}, \mathbf{d}_B^{(0)}, \cdots, \mathbf{d}_B^{(N-1)})$. Their concatenation $\mathbf{M}(C)$ is constructed by merging coefficients in $\mathbf{M}(A)$ and $\mathbf{M}(B)$. Coefficients of $\mathbf{M}(C)$ at the boundary of $\mathbf{M}(A)$ and $\mathbf{M}(B)$ are set to the average of the last coefficients of $\mathbf{M}(A)$ and the first coefficients of $\mathbf{M}(B)$. Letting the size of $\mathbf{m}_A^{(0)}$ be k_A, the number of coefficients in $\mathbf{d}_A^{(n)}$ is given as $k_A^n = 2^{n+1}(k_A - 1) + 1$. We can then formally describe $\mathbf{M}(C)$ as follows:

$$\begin{aligned}
\mathbf{m}_C^{(0)}(i) &= \begin{cases} \mathbf{m}_A^{(0)}(i), & \text{if } i < k_A, \\ \mathrm{lerp}_{\frac{1}{2}}(\mathbf{m}_A^{(0)}(k_A), \mathbf{m}_B^{(0)}(0)), & \text{if } i = k_A, \\ \mathbf{m}_B^{(0)}(i - k_A), & \text{if } i > k_A, \end{cases} \\
\mathbf{d}_C^{(n)}(i) &= \begin{cases} \mathbf{d}_A^{(n)}(i), & \text{if } i < k_A^n, \\ \frac{1}{2}(\mathbf{d}_A^{(n)}(k_A^n) + \mathbf{d}_B^{(n)}(0)), & \text{if } i = k_A^n, \\ \mathbf{d}_B^{(n)}(i - k_A), & \text{if } i > k_A^n, \end{cases}
\end{aligned} \tag{21}$$

where $0 \leq n < N$ and $\mathrm{lerp}_t(\mathbf{m}_A, \mathbf{m}_B)$ yields linear interpolation for translation components and spherical linear interpolation for rotation components. Since this algorithm stitches different levels of detail separately, the gross shape of the resulting motion signal is not seriously affected by small noise at the boundary of the input motion clips.

5.2. Coordinate Invariance

To show the importance of coordinate-invariance, it would be instructive to observe how a motion signal changes with multiresolution analysis by the choice of a reference frame. One very popular approach is to parameterize a motion signal with six parameters: three parameters for position components and the others for orientation components that are represented by Euler angles. With this parameterization, multiresolution analysis for orientation components can be done in the same way as was done for position components [1].

In our experiments, we first construct the multiresolution representation $M(\mathbf{m}) = (\mathbf{m}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(N-1)})$ of the motion signal and then transform it to reconstruct the signal at a different position in such a way that $M' = (\mathcal{T}\mathbf{m}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(N-1)})$. Then, we compare the reconstructed signal with $\mathcal{T}\mathbf{m}$ obtained by applying the same transformation directly to the original signal. In our experiments, the motion signal corresponding to the pelvis trajectory of "Jump and Kick" in Figure 6 is used to generate nine samples by rotating the original signal successively about the Y-axis (vertical) by the incremental angle of $\frac{\pi}{12}$ (Figure 10). Figure 11 shows the frame-by-frame difference between the transformed signal $\mathcal{T}\mathbf{m}$ and the signals reconstructed through multiresolution analysis. We measure the angular difference between two orientations by geodesic distance:

$$d(\mathbf{q}_1, \mathbf{q}_2) = \min(\|\log(\mathbf{q}_1^{-1}, \mathbf{q}_2)\|, \|\log(\mathbf{q}_1^{-1}, -\mathbf{q}_2)\|). \tag{22}$$

Here, since two antipodal points represent the same orientation, we must choose the minimum between $\|\log(\mathbf{q}_1^{-1}, \mathbf{q}_2)\|$ and $\|\log(\mathbf{q}_1^{-1}, -\mathbf{q}_2)\|$. As proved in the previous section, our method yields identical signals independent of their positions and orientations to be reconstructed. Unlike our method, our experiments show that the conventional method based on Euler angles generates quite different results depending on the transformation $\mathcal{T}$.

6. CONCLUSION

We have presented a new multiresolution approach to motion analysis and synthesis. Our motion representation allows us to modify the coefficients at each level in the hierarchy independently of those at the other levels through the level-wise manipulation of detail coefficients. Exploiting this capability, we have developed a variety of motion editing tools that can be used for modifying, blending, and stitching highly detailed motion data.

The success of our approach is mainly due to motion filtering and hierarchical displacement mapping. Our filtering scheme can handle orientations as well as positions in a coherent manner. The notion of hierarchical displacement mapping provides an elegant formulation for multiresolution representations in which each individual detail coefficient is represented as a pair of 3-dimensional vectors measured at a local coordinate system. This formulation leads to multiresolution motion synthesis through coordinate-independent operations.

ACKNOWLEDGEMENT

This work was supported in part by the NRL (National Research Laboratory) project of KISTEP (Korea Institute of Science & Technology Evaluation and Planning).

REFERENCES

1. A. Bruderlin and L. Williams. Motion signal processing. *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 97–104, August 1995.
2. P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2:215–236, 1983.
3. S. Dubuc. Interpolation through an iterative scheme. *J. Math. Anal. Appl.*, 114:185–204, 1986.
4. N. Dyn, J. Gregory, and D. Levin. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 7:129–140, 1987.
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. M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of Symposium on Interactive 3D Graphics*, pages 139–148, 1997.
7. M. Gleicher. Retargetting motion to new characters. *Computer Graphics (Proceedings of SIGGRAPH 98)*, pages 33–42, July 1998.
8. S. Guo and J. Robergé. A high-level control mechanism for human locomotion based on parametric frame space interpolation. In *Proceedings of Computer Animation and Simulation '96, Eurographics Animation Workshop*, pages 95–107. Springer-Verlag, 1996.
9. 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.
10. 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.
11. D. Kincaid and W. Cheney. *Numerical Analysis*. Books/Cole, 1990.
12. L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. *Computer Graphics (Proceedings of SIGGRAPH 98)*, pages 105–114, July 1998.

13. L. Kobbelt and P. Schröder. A multiresolution framework for variational subdivision. *ACM Transactions on Graphics*, 17(4):209–237, 1998.

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

15. 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.

16. J. Lee and S. Y. Shin. General construction of time-domain filters for orientation data. *IEEE Transactions on Computer Graphics and Visualization*, to appear.

17. Z. Liu, S. G. Gortler, and M. F. Cohen. Hierarchical spacetime control. *Computer Graphics (Proceedings of SIGGRAPH 94)*, pages 35–42, July 1994.

18. S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

19. C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and Adverbs: Multidimensional motion interpolation. *IEEE CG&A*, 18(5):32–40, October 1998.

20. C. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen. Efficient generation of motion transitions using spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 96)*, pages 147–154, August 1996.

21. P. Schröder, W. Sweldens, M. Cohen, T. DeRose, and D. Salesin. *Wavelets in Computer Graphics (SIGGRAPH 96 Course note #13)*. ACM press, 1996.

22. K. Shoemake. Animating rotation with quaternion curves. *Computer Graphics (Proceedings of SIGGRAPH 85)*, pages 245–254, 1985.

23. E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, 1994.

24. G. Taubin. A signal processing approach to fair surface design. *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 351–358, August 1995.

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

26. D. J. Wiley and J. K. Hahn. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE Virtual Reality Annual International Symposium '97*, pages 157–160. IEEE Computer Society Press, 1997.

27. A. Witkin and Z. Popović. Motion warping. *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 105–108, August 1995.

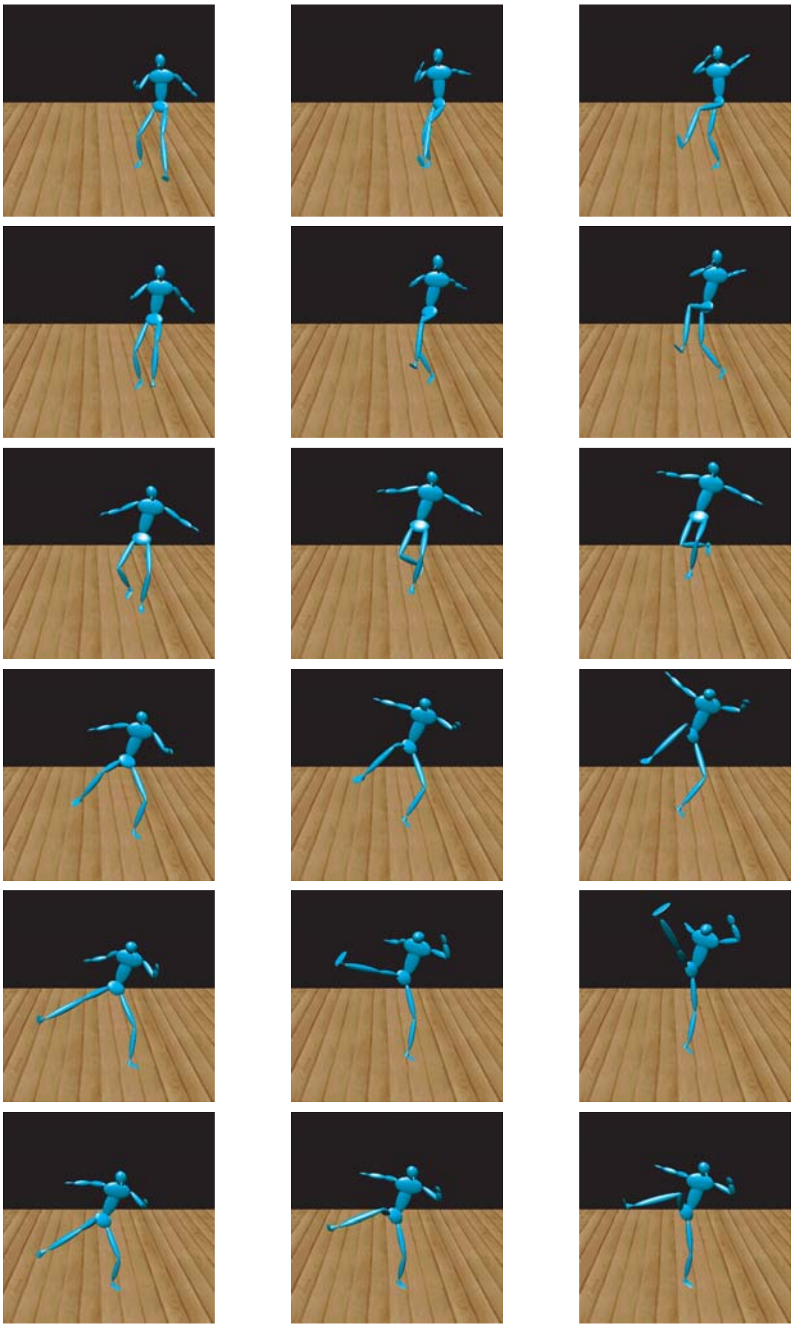

FIG. 6. Jump and kick. Attenuated (left); Original (center); Enhanced (right)

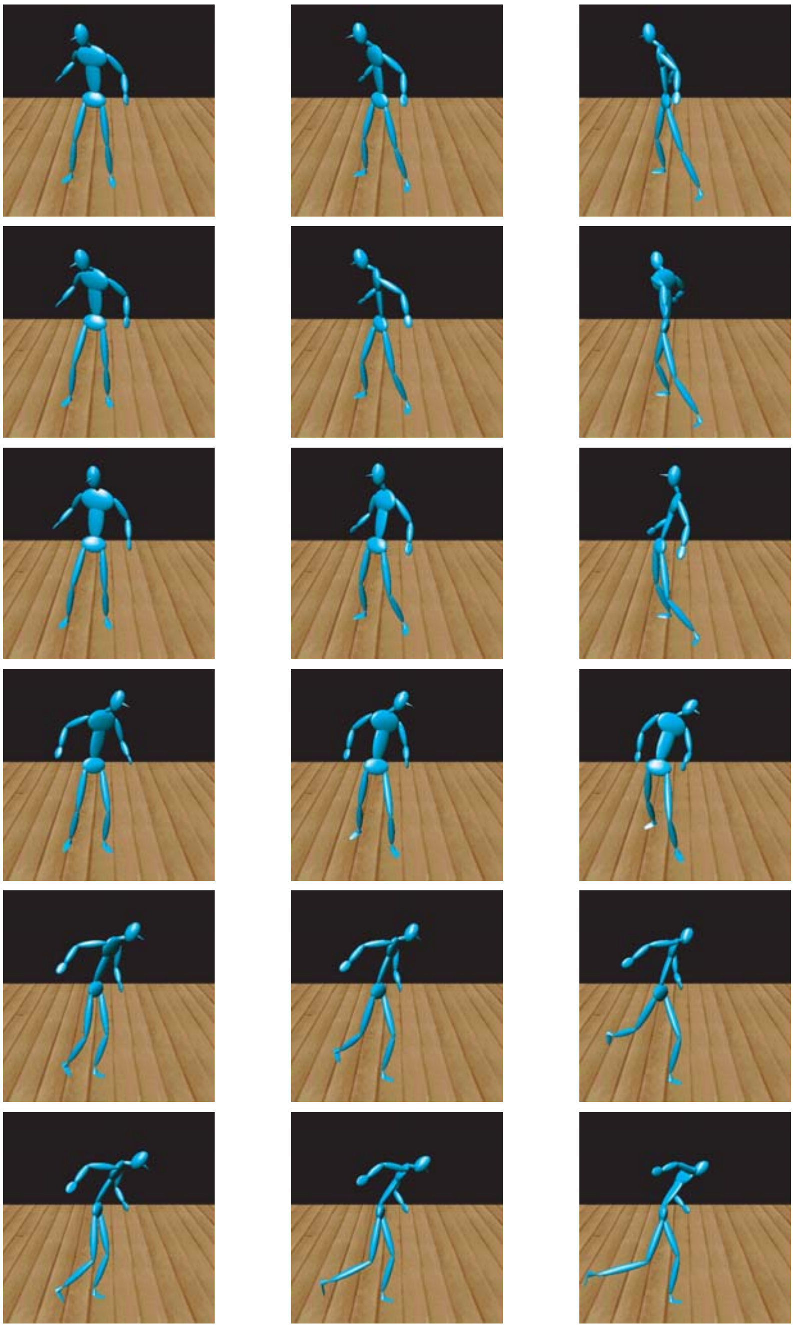

FIG. 7. Face hit. Attenuated (left); Original (center); Enhanced (right)

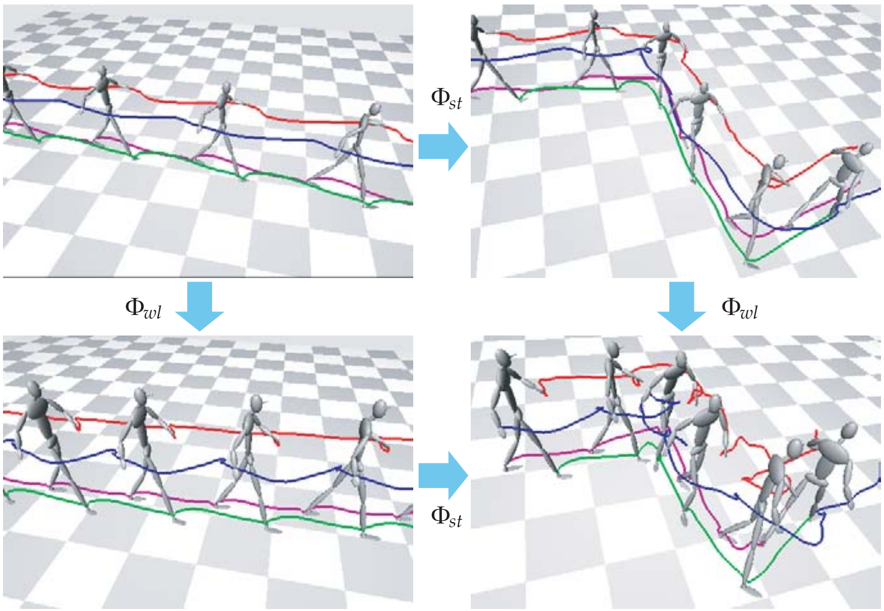

FIG. 8. Frequency-based motion blending. Straight walking (upper left); Turning with a normal walk (upper right); Walking with a limp (lower left); Turning with a limp (lower right)

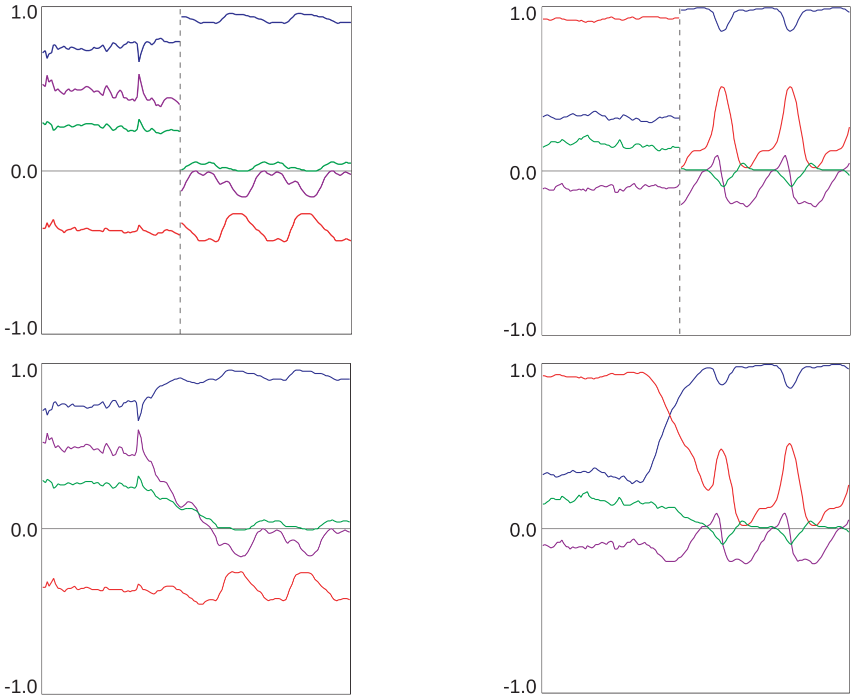

FIG. 9. Stitching live-captured motion clips. The left and right columns visualize the motion signals corresponding to the left elbow and right knee joints, respectively. Simple concatenation of the original signals yields a visual seam at the boundary (upper row); Level-wise stitching at the boundary connects the motion signals seamlessly (lower row).

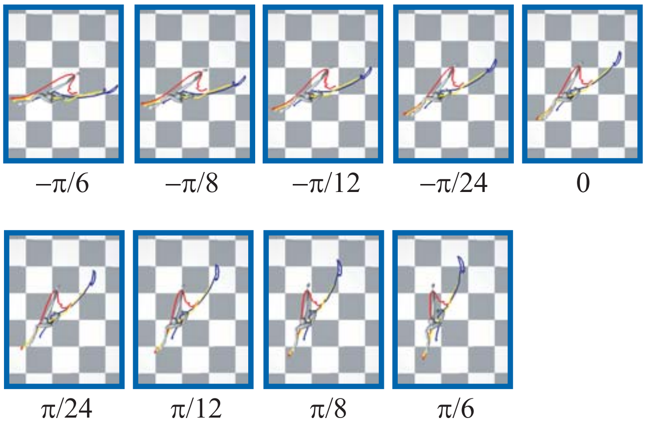

FIG. 10. The same motion clips placed at different orientations in the reference frame. The red, blue and yellow lines depict the trajectories of the left foot, right foot, and pelvis, respectively.

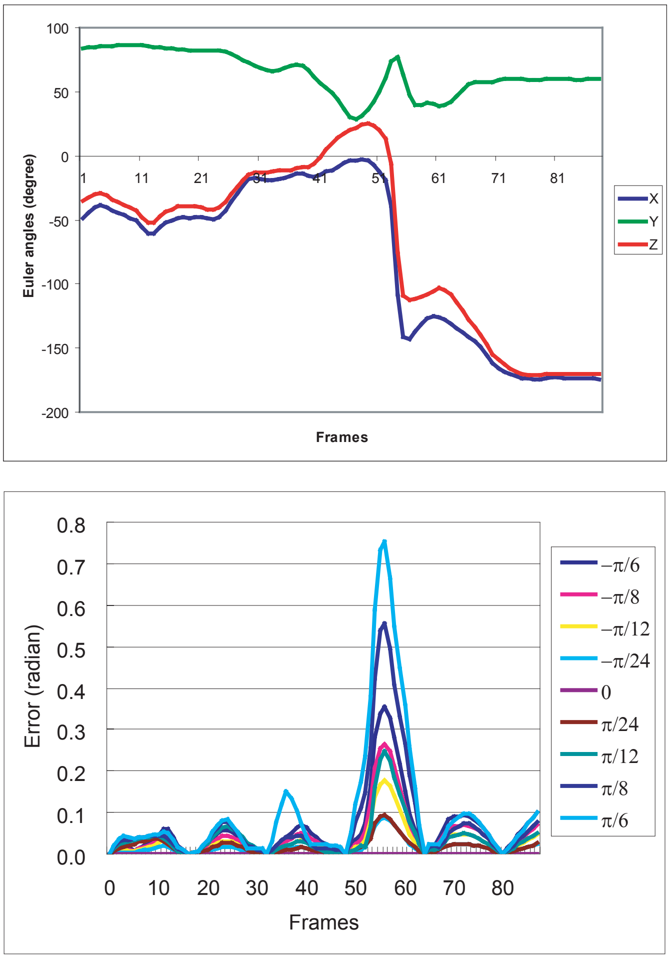

FIG. 11. Coordinate-dependence of Euler angles; The original trajectory of pelvis orientation (upper); The frame-by-frame differences between the original and the reconstructed signals (lower).