IIE Transactions (1998) **30**, 581–587

Noise smoothing for VR equipment in quaternions

C. C. HSIEH[1], Y. C. FANG[1], M. E. WANG[1], C. K. WANG[1], M. J. KIM[2], S. Y. SHIN[3] and T. C. WOO[4]*

[1]*Department of Industrial and Operations Engineering, University of Michigan, USA*
[2]*Computer Graphics Laboratory, System Engineering Research Institute, Taejon, Korea*
[3]*Department of Computer Science, Korea Advanced Institute of Science and Technology, Taejon, Korea*
[4]*Department of Industrial Engineering, Box 352650, University of Washington, Seattle, WA 98195-2650, USA*
E-mail: twoo@u.washington.edu

Received January 1997

Smooth motion generation is an important issue in the computer animation and virtual reality (VR) area. In general, the motion of a rigid body consists of translation and orientation. The former is described by a space curve in 3-dimensional Euclidean space, while the latter is represented by a curve in the unit quaternion space. Although there are well-known techniques for smoothing the translation curve in the Euclidean space, few results have been reported for smoothing motion as a whole. This paper improves the previous study and provides a more robust algorithm, which seeks to minimize the weighted sum of the strain-energy and the sum of the squared errors.

1. Introduction

Achieving smooth spatial motion is a challenging task in computer graphics with practical applications in the animation and virtual reality (VR) areas. The position, or, more precisely, the position of a reference point which is fixed with respect to the object, and orientation of a moving object at a given time are often paired and are called a "frame" of motion. Achieving smooth motion thus comprises two parts: the realization of smooth translation and that of smooth rotation. For the former, many curve-fitting and curve-fairing methods are known to provide varying degrees of continuity for different kinds of point data. As for the latter, until recent progress in *quaternions*, interpolating between orientations had been difficult.

Although rotations can be represented as a 3×3 matrix, directly interpolating corresponding entries is not viable due to the partial dependency of entries in a rotation matrix. In 1758, Euler [1] introduced a set of independent co-ordinates, called *Euler angles*, which describe a three-dimensional rotation as the composition of three independent, ordered rotations about given axes through an origin. Although interpolating rotations via Euler angles is feasible, it is cumbersome [2]. On the other hand, a quaternion represents an orientation as a single rotation, enabling simple interpolation schemes.

Recent advances in quaternion algebra have stimulated a considerable amount of interest in developing new techniques to achieve smooth motion. These techniques can be classified into two categories: motion synthesis [3–9] and motion fairing [10]. In motion synthesis, a series of "key frames" are first given, which, for example, may designate the crucial scenes, such as views of the kitchen, living room, etc., in a fly-through tour of a model house, then between these key frames, intermediate frames are interpolated using two spline curves, one in the Euclidean space R^3 and the other in the unit quaternion space $SO(3)$, to simulate smooth motion. Despite the success of motion-synthesis techniques in generating smooth animations such as fly-throughs and rigid-body motions, the automatic generation of smooth life-like animations for articulated creatures has yet to be achieved. The major factor for this unbalanced progress is that the motion of a moving creature is the result of complex interactions between muscles, bones, inertia, and the environment. Hence, in both animation and VR studies the physical world motions have to be sampled on many occasions before being reconstructed in the computer. In practice, rotational motion is usually sensed by gyroscopes which measure angular velocity, whilst translational motion can be determined using several techniques including telemetry analysis and linear accelerometers. As is the case for most sampled data, the successful reconstruction of sampled motions depends significantly on the ability to reduce noise during the sampling process. Motion fairing addresses this issue of reducing the jerkiness of sampled motions.

The use of curve-fairing techniques in R^3 is an area that has been extensively studied [11,12] and they have been shown to provide varying degrees of continuity for dif-

*To whom correspondence should be addressed.

ferent kinds of point data. However very few results have been reported for smoothing motion as a whole with the notable exception of the work of Lee and Shin [10]. They presented a fairing algorithm which is an extension of the strain-energy minimizing technique in R^3 discussed by Eck and Jaspert [13]. This algorithm iteratively minimizes the weighted sum of the derivatives of the unit quaternion curve, which describes the strain energy stored in the curve, within a user-specified error bound from the original data. Two areas in the algorithm in Lee and Shin [10] require improvement, namely: (i) the formulation is scale-dependent, i.e., the finally obtained motion varies with the scale of the translational data; and (2) under some conditions, the algorithm can loop infinitely. To amend these two shortcomings, we propose a more robust formulation in this paper, which seeks to minimize the weighted sum of the strain-energy and the sum of squared errors (SSE). Our formulation is less prone to infinite loops and gives the user an intuitive control over the smoothness of the final motion by adjusting the relative weight of the strain energy to the SSE. Also, our formulation is invariant over uniform scaling of the position data.

The rest of this paper is organized as follows: Section 2 gives a review of the crucial properties of a unit quaternion, Section 3 presents our motion-fairing technique with experimental results, and Section 4 concludes with a brief summary.

2. Preliminary

A *quaternion* $\boldsymbol{a}$, in the form of a complex number, consists of one real part and three imaginary parts:

$$\boldsymbol{a} = w + x\hat{i} + y\hat{j} + z\hat{k}, \quad w,x,y,z \in R. \tag{1}$$

which satisfies the multiplication rules [14,15]:

$$\begin{gathered}\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = -1,\\ \hat{i}\hat{j} = \hat{k}, \;\; \hat{j}\hat{i} = -\hat{k},\\ \hat{j}\hat{k} = \hat{i}, \;\; \hat{k}\hat{j} = -\hat{i},\\ \hat{k}\hat{i} = \hat{j}, \;\; \hat{i}\hat{k} = -\hat{j}.\end{gathered}$$

The norm $||\boldsymbol{a}||$ and the inverse $\boldsymbol{a}^{-1}$ of a quaternion $\boldsymbol{a}$ are defined as follows:

$$\begin{aligned}||\boldsymbol{a}|| &= \sqrt{w^2 + x^2 + y^2 + z^2},\\ \boldsymbol{a}^{-1} &= (w - x\hat{i} - y\hat{j} - z\hat{k})/||\boldsymbol{a}||.\end{aligned}$$

A quaternion $\boldsymbol{a}$ can also be represented as an ordered pair $\boldsymbol{a} = (w, \vec{u}), \vec{u} = (x,y,z) \in R^3$. A vector $\vec{\boldsymbol{v}} \in R^3$ can thus be easily mapped into the quaternion space:

$$\vec{\boldsymbol{v}} \longrightarrow (0, \vec{\boldsymbol{v}}). \tag{2}$$

A unit quaternion $\boldsymbol{q}$ is a quaternion of the norm $||\boldsymbol{q}|| = 1$. By using a unit quaternion as an alternative to 3×3 rotation matrices, the rotation by θ about a unit axis vector $\hat{u}$ can be represented as a unit quaternion $\boldsymbol{q}$:

$$\boldsymbol{q} = (\cos\frac{1}{2}\theta, \hat{u}\sin\frac{1}{2}\theta), \tag{3}$$

and then a vector $\vec{\boldsymbol{v}}_1 \in R^3$ rotated from $\vec{\boldsymbol{v}} \in R^3$ by the rotation $\boldsymbol{q}$, denoted by $R_{\boldsymbol{q}}(\vec{\boldsymbol{v}})$, can be expressed as a quaternion multiplication.

$$\begin{aligned}\vec{\boldsymbol{v}}_1 &= R_{\boldsymbol{q}}(\vec{\boldsymbol{v}}),\\ &= \boldsymbol{q}\vec{\boldsymbol{v}}\boldsymbol{q}^{-1},\end{aligned} \tag{4}$$

where $\vec{\boldsymbol{v}}_1$ and $\vec{\boldsymbol{v}}$ follow (2). The generalization of (4) in which successive rotations $R_{\boldsymbol{q}_1}, \ldots, R_{\boldsymbol{q}_n}$ from $\vec{\boldsymbol{v}}$ take place is given by:

$$\begin{aligned}R_{\boldsymbol{q}_n}(\cdots R_{\boldsymbol{q}_2}(R_{\boldsymbol{q}_1}(\vec{\boldsymbol{v}}))\cdots) &= \boldsymbol{q}_n \cdots \boldsymbol{q}_1 \vec{\boldsymbol{v}} \boldsymbol{q}_1^{-1} \cdots \boldsymbol{q}_n^{-1},\\ &= (\boldsymbol{q}_n \cdots \boldsymbol{q}_1)\vec{\boldsymbol{v}}(\boldsymbol{q}_1^{-1} \cdots \boldsymbol{q}_n^{-1}),\\ &= R_{\boldsymbol{q}_n \cdots \boldsymbol{q}_1}(\vec{\boldsymbol{v}}).\end{aligned}$$

The well-known formation of an angular velocity $\vec{\boldsymbol{v}} = \theta\vec{\boldsymbol{u}} \in R^3$ in classical mechanics is represented by a rotation $\theta = ||\vec{\boldsymbol{v}}||$ about the unit axis $\hat{u} = \vec{\boldsymbol{v}}/||\vec{\boldsymbol{v}}||$. The *exponential* map from $\vec{\boldsymbol{v}}$ to its corresponding unit quaternion is defined as [15]:

$$\begin{aligned}\exp(\vec{\boldsymbol{v}}) &= (\cos\theta, \hat{u}\sin\theta),\\ &= \left(\cos\frac{||\vec{\boldsymbol{v}}||}{2}, \frac{\vec{\boldsymbol{v}}}{||\vec{\boldsymbol{v}}||}\sin\frac{||\vec{\boldsymbol{v}}||}{2}\right).\end{aligned} \tag{5}$$

As the quaternion exponential (5) is not one-to-one, the constraint $||\vec{\boldsymbol{v}}|| < \pi$ is imposed so that the inverse function $\log \boldsymbol{q}$ of a unit quaternion $\boldsymbol{q}$ is well defined.

Given two unit quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, it is possible to form a path connecting them by a rotation with a fixed axis. One such rotation path is the *geodesic* interpolation or *slerp* [2] which is a result of the Euler principal rotation theorem [15]:

$$\begin{aligned}G[\boldsymbol{q}_1,\boldsymbol{q}_2](t) &= \boldsymbol{q}_1 \exp(t\log(\boldsymbol{q}_1^{-1}\boldsymbol{q}_2)), \quad 0 \le t \le 1,\\ &= \boldsymbol{q}_1(\boldsymbol{q}_1^{-1}\boldsymbol{q}_2)^t, \quad 0 \le t \le 1.\end{aligned} \tag{6}$$

where $\log(\boldsymbol{q}_1^{-1}\boldsymbol{q}_2)$ is the corresponding rotation axis. The geodesic path is also embedded in the distance metric for the unit quaternion space so that the distance between two unit quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is defined by :

$$\mathrm{dist}(\boldsymbol{q}_1, \boldsymbol{q}_2) = ||\log(\boldsymbol{q}_1^{-1}\boldsymbol{q}_2)||. \tag{7}$$

Unlike in the Euclidean space in which the velocity $v(t)$ of a space curve $\boldsymbol{p}(t)$ is the first derivative $\boldsymbol{p}'(t)$, the angular velocity $\omega(t)$ of a unit quaternion curve $\boldsymbol{q}(t)$ takes a nonlinear form [16,17]:

$$\omega(t) = 2\boldsymbol{q}^{-1}(t)\boldsymbol{q}'(t). \tag{8}$$

On the other hand, given the angular velocity $\omega(t)$, the corresponding unit quaternion curve $\boldsymbol{q}(t)$ can be derived by the quaternion integration [18]:

$$\boldsymbol{q}(t) = \boldsymbol{q}_0 \prod_0^t \exp(\omega(t)\mathrm{d}t). \tag{9}$$

where $\boldsymbol{q}_0$ is the initial orientation.

3. Energy and distance measures

In this section, a set of rigid postures is smoothed in terms of energy and distance minimization. The discussion begins with the formulation of the energy measure for a space curve in R^3, followed by that for a unit quaternion curve. Together with the distance measures, the objective functions are established and the numerical method for solving the minimization problem is then given, with numerical results demonstrated.

3.1. *Energy functions*

The motion $\boldsymbol{r}(t)$ of a rigid body can be represented as the combination of the position and the orientation. The former is described by a space curve $\boldsymbol{p}(t)$ in R^3, while the latter by a unit quaternion curve $\boldsymbol{q}(t)$. Thus:

$$\boldsymbol{r}(t) = (\boldsymbol{p}(t), \boldsymbol{q}(t)).$$

Let $\boldsymbol{p}$ be a space curve in R^3 parameterized by time t. Then velocity $v(t)$ and the acceleration $a(t)$ of the curve $\boldsymbol{p}$ at t are:

$$v(t) = \boldsymbol{p}'(t), \tag{10}$$

$$a(t) = \boldsymbol{p}''(t). \tag{11}$$

As the acceleration (11) measures the rate of change of the neighboring velocities at t, the smoothness of a curve can be evaluated by minimizing the total acceleration:

$$\begin{aligned} E(\boldsymbol{p}) &= \int_{\boldsymbol{p}} ||\boldsymbol{p}''(t)||^2 \, \mathrm{d}t, \\ &= \int_{\boldsymbol{p}} ||a(t)||^2 \, \mathrm{d}t. \end{aligned} \tag{12}$$

The smaller $E(\boldsymbol{p})$ is, the smoother the curve $\boldsymbol{p}$. One such smooth curve, though a degenerate solution, is a straight line which has $E = 0$, since the acceleration $a(t)$ is zero everywhere. Note that if $\boldsymbol{p}$ is parameterized by an arc length s, then Equation (12) represents the strain energy of the curve $\boldsymbol{p}(s)$:

$$\begin{aligned} E(\boldsymbol{p}) &= \int_{\boldsymbol{p}} |\boldsymbol{p}''(s)|^2 \mathrm{d}s \\ &= \int_{\boldsymbol{p}} \kappa(s)^2 \mathrm{d}s. \end{aligned}$$

By analogy, it is reasonable to measure the smoothness of a unit quaternion curve $\boldsymbol{q}(t)$ in terms of the rate of change of angular velocities. Of note is that the first derivative of a unit quaternion curve is not the angular velocity, nor is the second derivative of the quaternion curve the rate of change of angular velocities. The energy function $\int_{\boldsymbol{q}} ||\boldsymbol{q}''(t)||^2 \mathrm{d}t$, which is directly derived from (12), is therefore not a good measure [10,19]. In Lee and Shin [10], it is also shown that a smooth unit quaternion curve of a constant angular velocity, passing through a great circle of the unit hyper-sphere, does not meet the measure $\int_{\boldsymbol{q}} ||\boldsymbol{q}''(t)||^2 \mathrm{d}t$. A more appropriate energy function can be obtained by directly integrating the norm of the derivative of the angular velocity:

$$E(\boldsymbol{q}) = \int_{\boldsymbol{q}} ||\omega'(t)||^2 \mathrm{d}t. \tag{13}$$

It is clear that the energy function $E(\boldsymbol{q}) = 0$ for a unit quaternion curve with a constant angular velocity, passing through the great circle of the unit hyper-sphere.

In practice, the motion is described as a pair of positions and orientations $(\boldsymbol{P}, \boldsymbol{Q})$ where $\boldsymbol{P} = \{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}\}$, $\boldsymbol{p}_i \in R^3$, is a sequence of positions and $\boldsymbol{Q} = \{\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{n-1}\}$ a sequence of unit quaternions, describing the orientations. Thus discrete energy functions are necessary. For a discrete positional curve $\boldsymbol{P} = \{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}\}$ in R^3, the first and second derivatives are given by:

$$\Delta^{(1)}\boldsymbol{p}_i = \frac{\boldsymbol{p}_{i+1} - \boldsymbol{p}_{i-1}}{2h}, \tag{14}$$

$$\Delta^{(2)}\boldsymbol{p}_i = \frac{\boldsymbol{p}_{i+1} - 2\boldsymbol{p}_i + \boldsymbol{p}_{i-1}}{h^2}, \tag{15}$$

where h is the time interval between two successive points $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}, i \in [0..n-2]$. Then the discrete energy function of $\boldsymbol{P}$ is given by:

$$\begin{aligned} E(\boldsymbol{P}) &= \sum_{i=0}^{n-2} ||\Delta^{(2)}\boldsymbol{p}_i||^2, \\ &= \sum_{i=0}^{n-2} \left|\left| \frac{\boldsymbol{p}_{i+1} - 2\boldsymbol{p}_i + \boldsymbol{p}_{i-1}}{h^2} \right|\right|^2. \end{aligned} \tag{16}$$

For a discrete quaternion curve $\boldsymbol{Q} = \{\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{n-1}\}$, on the other hand, the discrete angular velocity ω_i and the derivative ω_i' at $\boldsymbol{q}_i$ are given by:

$$\omega_i = \frac{\log(\boldsymbol{q}_i^{-1}\boldsymbol{q}_{i+1})}{h}, \tag{17}$$

$$\omega_i' = \frac{\log(\boldsymbol{q}_i^{-1}\boldsymbol{q}_{i+1}) - \log(\boldsymbol{q}_{i-1}^{-1}\boldsymbol{q}_i)}{h^2}, \tag{18}$$

and then the discrete energy function of $\boldsymbol{Q}$ is given by:

$$\begin{aligned} E(\boldsymbol{Q}) &= \sum_{i=0}^{n-2} ||\omega_i'||^2, \\ &= \sum_{i=0}^{n-2} \left|\left| \frac{\log(\boldsymbol{q}_i^{-1}\boldsymbol{q}_{i+1}) - \log(\boldsymbol{q}_{i-1}^{-1}\boldsymbol{q}_i)}{h^2} \right|\right|^2. \end{aligned} \tag{19}$$

3.2. *Distance measure*

The energy functions (16) and (19) are necessary yet not sufficient to smooth either a positional or a rotational data set. This is because by only minimizing the energy functions, the following problems can occur: (1) trivial solutions are likely to occur; (2) the positional data points

can be pulled to form a straight line; and (3) the quaternion points can be relocated to form a great arc. In order to preserve the shape of the data sets, an L_2-*norm* metric is thus taken into account which measures the sum of the squared distances of the corresponding data points in the original data set and those in the smoothed data set. Let $\boldsymbol{P}^* = \{\boldsymbol{p}_0^*, \boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_{n-1}^*\}$ and $\boldsymbol{Q}^* = \{\boldsymbol{q}_0^*, \boldsymbol{q}_1^*, \ldots, \boldsymbol{q}_{n-1}^*\}$ be the resulting data sets of $\boldsymbol{P}$ and $\boldsymbol{Q}$ after the smoothing process respectively. Then, the distance measures $D(\boldsymbol{P}^*)$ and $D(\boldsymbol{Q}^*)$ are:

$$D(\boldsymbol{P}^*) = \sum_{i=0}^{n-1} ||\boldsymbol{p}_i - \boldsymbol{p}_i^*||^2, \tag{20}$$

$$D(\boldsymbol{Q}^*) = \sum_{i=0}^{n-1} ||\log(\boldsymbol{q}_i^{-1}\boldsymbol{q}_i^*)||^2. \tag{21}$$

Clearly, the distance measures (20) and (21) are minimized when the smoothed data sets are identical to the original data sets.

3.3. *Optimization and its numerical method*

The formulations for smoothing the positional data set $\boldsymbol{P}$ and that for the orientation data set $\boldsymbol{Q}$ are now ready to be established. The solution $\boldsymbol{P}^*$ for smoothing the positional data set can be obtained by minimizing the objective function $F(\boldsymbol{P}^*)$:

$$\min F(\boldsymbol{P}^*) = D(\boldsymbol{P}^*) + \alpha E(\boldsymbol{P}^*), \tag{22}$$

where $D(\boldsymbol{P}^*)$ and $E(\boldsymbol{P}^*)$ follow (16) and (20) respectively; and the constant α is the ratio of the significance of the energy measure over that of the distance measure. Clearly, if $\alpha = 0$, the solution for (22) would solely be the set $\boldsymbol{P}$. If $\alpha \gg 0$, the energy function becomes dominant thus the solution $\boldsymbol{P}^*$ would be the straightened version of $\boldsymbol{P}$. Likewise, the solution $\boldsymbol{Q}^*$ for smoothing the quaternion data can be obtained by minimizing the objective function $G(\boldsymbol{Q}^*)$:

$$\min G(\boldsymbol{Q}^*) = D(\boldsymbol{Q}^*) + \beta E(\boldsymbol{Q}^*), \tag{23}$$

where $D(\boldsymbol{Q}^*)$ and $E(\boldsymbol{Q}^*)$ follow (19) and (21) respectively; the constant β again is the ratio of the significance of the energy measure over that of the distance measure.

The numerical method for solving (22) consists of three steps:

Step 1. Initialization:

$$\boldsymbol{P}^0 \longleftarrow \boldsymbol{P}.$$

Step 2. Iteration:

$$\boldsymbol{P}^{k+1} = \boldsymbol{P}^k - \lambda_1 \nabla F(\boldsymbol{P}^k).$$

Step 3. Stopping criterion:

$$||\nabla F(\boldsymbol{P}^k)|| < \varepsilon_1.$$

Step 1 simply specifies the initial condition for the iterative point $\boldsymbol{P}^0$. In step 2, a sequence of $\{\boldsymbol{P}^k\}$ is generated by the iteration $\boldsymbol{P}^{k+1} = \boldsymbol{P}^k - \lambda_1 \nabla F(\boldsymbol{P}^k)$, where $\nabla F(\boldsymbol{P}^k)$ is the gradient of $F(\boldsymbol{P}^k)$ and the small number λ_1 determines the convergence speed. The iteration will terminate when the gradient $\nabla F(\boldsymbol{P}^k)$ of the current iteration is bounded by a pre-specified constant ε_1, i.e., the current iterative point set $\boldsymbol{P}^k$ is close to the local minimum. Thus, the solution $\boldsymbol{P}^*$ would be $\boldsymbol{P}^* = \boldsymbol{P}^k$.

Similarly, the numerical procedure for solving (23) is as follows:

Step 1. Initialization:

$$\boldsymbol{Q}^0 \longleftarrow \boldsymbol{Q}.$$

Step 2. Iteration:

$$\boldsymbol{Q}^{k+1} = \boldsymbol{Q}^k \exp(-\lambda_2 \nabla G(\boldsymbol{Q}^k)).$$

Step 3. Stopping criterion:

$$||\nabla G(\boldsymbol{Q}^k)|| < \varepsilon_2.$$

The initialization of the starting point $\boldsymbol{Q}^0$ and the termination of the iteration are described in Step 1 and Step 3 respectively. In Step 2, the iteration, which is different from that for solving (22), occurs. We will describe this procedure in the next paragraph.

By analogy to the previous numerical procedure, the iteration would be $\boldsymbol{Q}^{k+1} = \boldsymbol{Q}^k - \lambda_2 \nabla G(\boldsymbol{Q}^k)$. However, since this iteration would not generate unit quaternion points in general, a better iteration scheme is required. As the neighboring unit quaternion points of the unit quaternion $\boldsymbol{q}_i$ can be represented as $\boldsymbol{q}_i \exp(\omega)$ where ω is a 3-dimensional rotation vector, the component $\nabla_i G(\boldsymbol{Q})$ of the gradient $\nabla G(\boldsymbol{Q})$ for $\boldsymbol{q}_i$ can then be defined by:

$$\nabla_i G(\boldsymbol{Q}) = \frac{\partial}{\partial \omega}\bigg|_{\omega=0} G(\boldsymbol{q}_0, \boldsymbol{q}_0, \ldots, \boldsymbol{q}_i \exp(w), \ldots, \boldsymbol{q}_{n-1}), \tag{24}$$

and thus the gradient $\nabla G(\boldsymbol{Q})$ is:

$$\nabla G(\boldsymbol{Q}) = \{\nabla_i G(\boldsymbol{Q})\}. \tag{25}$$

The derivation of $\nabla G(\boldsymbol{Q})$ which requires the differentiation of the quaternion multiplication, exponential, and logarithm can be found in Kim *et al*. [16]. Now, the iterative points can be defined in a manner similar to that in the previous procedure:

$$\boldsymbol{Q}^{k+1} = \boldsymbol{Q}^k \exp(-\lambda_2 \nabla G(\boldsymbol{Q}^k)),$$

which follows (24) and (25).

3.4. *Numerical results*

The motion data $\boldsymbol{R} = (\boldsymbol{P}, \boldsymbol{Q})$ are sampled from a perturbed B-spline curve and a perturbed quaternion B-spline curve, both of which are injected with random

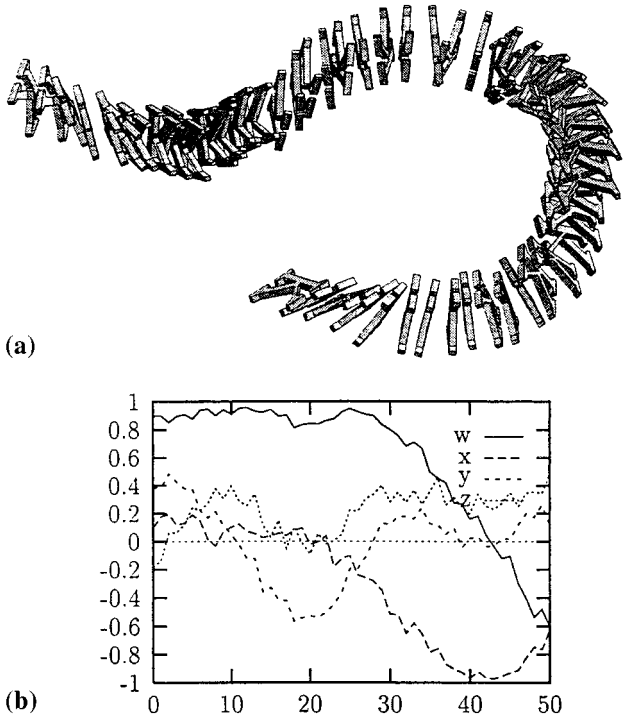

Fig. 1. (a) The motion of a rigid body and (b) the four components of the associated quaternions.

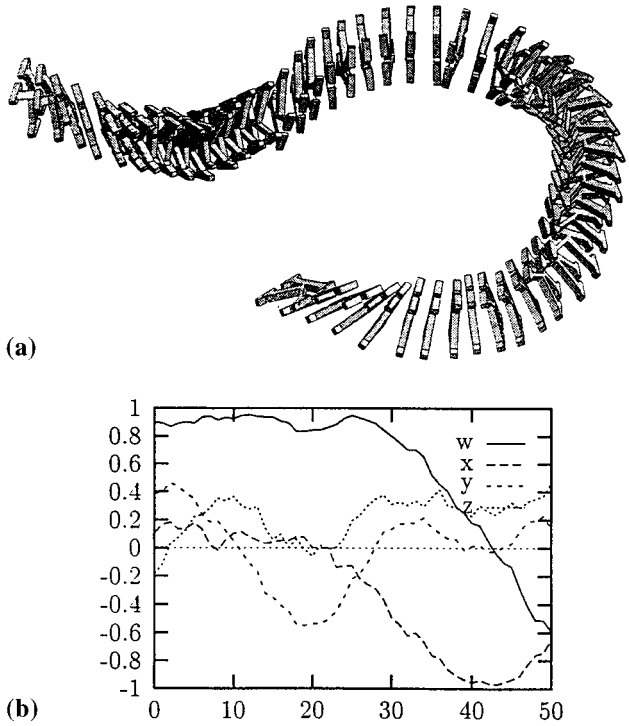

Fig. 3. (a) The smoothed motion and (b) the smoothed components of the quaternions produced when $\alpha = \beta = 0.1$.

noise. Note that the quaternion data are perturbed with random direction and random geodesic length on the unit hyper-sphere so that the perturbed points are guaranteed to be unit quaternions. To be more illustrative, relatively dense samples are taken with respect to the magnitude of the random noise. As is shown in Fig. 1 (a and b), the sequence of the positions for both the motion and orientations are not smooth. By applying the numerical method described in the preceding subsection with a small α value of 0.7, the resulting positions of the motion, that resemble the original trajectory of the motion, appears to be smoother, as is shown in Fig. 2. The minimization of the orientations with $\beta = 0.1$, along with the whole smoothed motion is shown in Fig. 3(a), while the smoothed quaternions are shown in Fig. 3(b). It is noticeable in Fig. 3 (a and b) that disorientation exists along the motion path. As discussed earlier, the smoothness of the orientations can be increased by the use of larger values of β. Figures 4 and 5 (a and b), illustrate the smoothed positions and the smoothed motion for $\alpha = \beta = 0.3$. It is clear that in Fig. 5 (a and b), the orientation of the motion appears smoother than that in Fig. 3 (a and b).

4. Summary

A minimization method for smoothing motion data has been established. The smoothness of the motion data can be controlled by the factors α and β, each of which is the

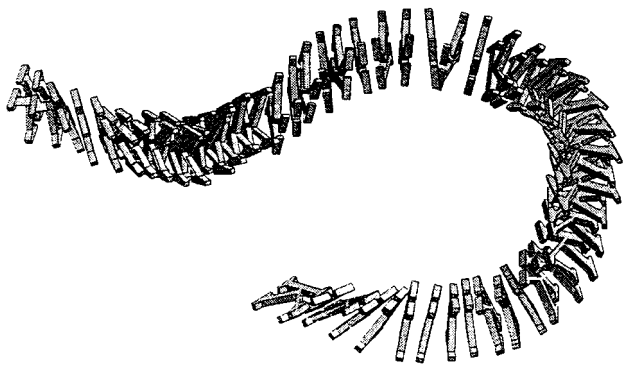

Fig. 2. The resulting positions of the motion where $\alpha = 0.1$.

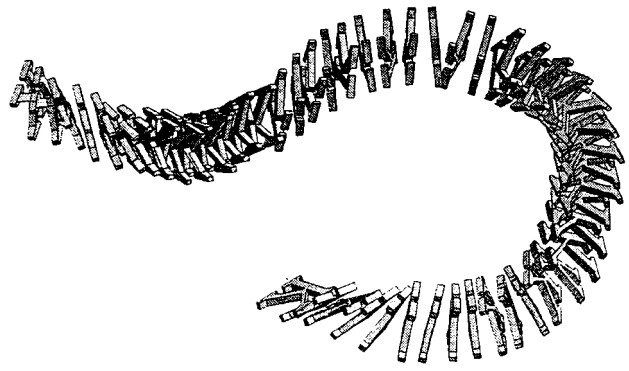

Fig. 4. The resulting positional curve where $\alpha = 0.3$.

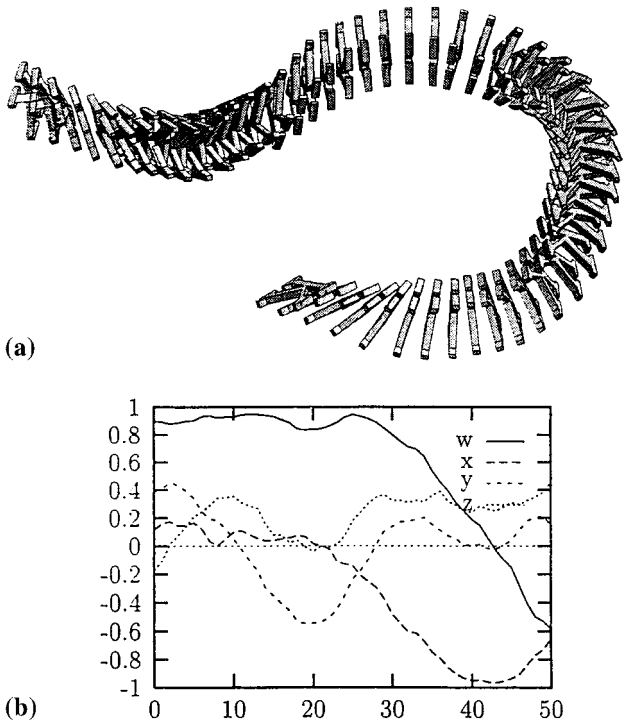

Fig. 5. (a) the smoothed motion and (b) the smoothed quaternions produced when $\alpha = \beta = 0.3$.

ratio of the significance of the corresponding energy measure to that of the corresponding distance measure. For $\alpha = \beta = 0$, the smoothed motion is identical to the original motion. As α and β are increased, a smoother motion can be achieved without the loss of the original shape of the motion. The examples presented in this paper also show the robustness of our algorithm.

References

[1] Euler, L. (1758) Du movement de rotation des corps solides autour d'un axe variable, in *Opera Omnia, Ser. Secunda*, Vol 8, Orell Füsli Turiei, Lausanne.

[2] Shoemake, K. (1985) Animating rotation with quaternion curves, in *Proceedings of SIGGRAPH '85*, Vol. **19**, pp. 245–254.

[3] Jin, X., Bao, H. and Peng, Q. (1994) Angular velocity interpolation using quaternion, in *Proceedings of the 4th International Conference on Computer-Aided Drafting, Design and Manufacturing Technology*, Vol. 1, pp. 25–30.

[4] Johnstone, J.K. and Williams, J.P. (1995) Rational control of orientation for animation, in *Proceedings of Graphics Interface '95*, pp. 179–186.

[5] Kim, M.J., Kim, M.S. and Shin, S.Y. (1995) A C^2-continuous B-spline quaternion curve interpolating a given sequence of solid orientations, in *Proceedings Computer Animation '95*, pp. 72–81.

[6] Kim, M.S. and Nam, K.W. (1993) Interpolating solid orientations with circular blending quaternion curves, in *Proceedings of the Communicating with Virtual Worlds*, pp. 258–269.

[7] Lake, R. and Green, M. (1991) Dynamic motion control of an articulated figure using quaternion curves, in *Proceedings of the Second International Conference on Computer-Aided Design and Computer Graphics*, pp. 37–44.

[8] Nielson, G.M. and Heiland, R.W. (1992) Animated rotations using quaternions and splines on a 4D sphere. *Programming and Computer Software*,**18**, 145–154.

[9] Wang, W. and Joe, B. (1993) Orientation interpolation in quaternion space using spherical biarcs, in *Proceedings Graphics Interface '93*, pp. 24–32.

[10] Lee, J. and Shin, S.Y. (1996) Motion fairing, in *Proceedings of Computer Animation*.

[11] Choi, B.K. (1991) *Geometric Modeling for CAD/CAM*, Elsevier Science Publishing, New York, NY.

[12] Farin, G. (1990) *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 2nd edn. Academic Press, New York, NY.

[13] Eck, M. and Jaspert, R. (1994) Automatic fairing of point sets, in *Designing Fair Curves and Surfaces*, N.S. Sapidis (ed), pp. 45–60.

[14] Curtis, M. (1972) *Matrix Groups* Springer-Verlag.

[15] Hamilton, W.R. (1969) *Elements of Quaternions* Vol. I, II, Chelsea Publishing Company.

[16] Kim, M.J., Kim, M.S. and Shin, S.Y. (1996) A compact differential formula for the first derivative of a unit quaternion curve. *Journal of Visualization and Computer Animation*,**7**, 43–57.

[17] Junkins, J.K. and Turner, J.D. (1980) Optional continuous torque attitude maneuvers. *Journal of Guidance and Control*, **3**, 210–217.

[18] Kim, M.J. (1996) General schemes for unit quaternion curve construction. Ph.D. thesis, KAIST, Taejon, Korea.

[19] Barr, A., Currin, B., Gabriel, S. and Hughes J. (1992) Smooth interpolation of orientations with angular velocity constraints, in *Proceedings of SIGGRAPH 92*, Vol. 26, pp. 313–320.

Biographies

Chung-Chi Hsieh is an Assistant Professor at ChaoYang University of Technology. He received a BS in Industrial Engineering and a BS in Electrical Engineering from National Tsing Hua University (Taiwan) in 1990. He completed his Ph.D. in Industrial Operations Engineering at the University of Michigan in 1997. His research interests are in computational geometry, computer graphics, tolerancing, and inspection in manufacturing processes.

Ying-Che Fang is currently a Ph.D. candidate in the Department of Industrial and Operations Engineering at the University of Michigan. He received his BS degree in Civil Engineering and Management Science from the National Chiao Tung University at Hsinshu, Taiwan in 1989 and an MS degree in Manufacturing Engineering from Syracuse University in 1993. His primary research interests are in the areas of CAD/CAM computational geometry, and geometric modeling.

Ming-En Wang received his B.S. in Mechanical Engineering from the National Chiao-Tung University in Taiwan and an M.S. in Industrial Engineering from Iowa State University. He is presently a Ph.D. candidate in Industrial and Operations Engineering at the University of Michigan. He joined Ford Motor Company as an Application Engineer in 1996, developing software tools for the processing and die-design of car-body stampings. His research interests include geometric problems in design and manufacturing, computational geometry, and computer graphics.

Chi-Kuo Gregory Wang is a Ph.D candidate in Industrial and Operations Research at the University of Michigan. He is currently participating in the Product Information Management project for die design at the Ford Motor Company. His research interests include CAD/CAM, computer graphics and computational geometry such as offset curves, visibility in reflective environment, Non-uniform B-spline curves in quaternion space and its application in Virtual Reality.

Chi-Kuo Gregory Wang received an M.Sc. degree in Mechanical Engineering in 1993 from the Ohio State University and a B.Sc. degree in Physics in 1989 from the National Tsing-Hua University.

Myoung-Jun Kim is a member of the Computer Graphics Laboratory, Systems Engineering Research Institute, in Taejon, Korea. After receiving his Ph.D. in Computer Science from the Korea Advanced Institute of Science and Technology in 1996, he spent a year at the University of Washington. There, he was instrumental in two projects: Beaconing Bar Code and Virtual Holography. Dr. Kim has broad interests beyond quarternions, geometry, and graphics. He has extensive industrial experience with tele-communications software. He can operate machine tools for mold and die making. His hobbies include musical instruments in particular synthesizers and boating.

Sung-Yong Shin is professor of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST) in Taejon, Korea. Before receiving his Ph.D from the University of Michigan in 1986, he directed software development at Samsung Data Systems. At KAIST, he teaches computer graphics and computational geometry; he also directs a research group in the area. He has served extensively the computer graphics community, in particular, the Pacific Graphics and the Computer Graphics International societies.

Tony Woo is a Professor of Industrial Engineering, Adjunct Professor of Mechanical Engineering, and holder of the John M. Fluke Distinguished Chair in Manufacturing Engineering at the University of Washington. He previously held positions at the University of Michigan and the National Science Foundation. His research and teaching interests are in computational geometry for design and manufacturing. He received his education in Electrical Engineering at the University of Illinois.