

Rational control of orientation for animation *

John K. Johnstone
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, Alabama, USA 35294
johnstone@cis.uab.edu
Tel. 205-934-2213

James P. Williams
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland, USA 21218
jimbo@cs.jhu.edu
Tel. 410-516-5298

Abstract

The smooth interpolation of keyframes of a rigid body, in particular their orientations, is an important problem in animation. Using the quaternion as a representation for orientation, several papers have solved this problem through the generation of smooth curves on the quaternion sphere. However, none of these methods have constructed rational curves.

This paper develops a method for generating true *rational Bezier* curves on the quaternion sphere that interpolate a given set of orientations. The control of orientation by a rational Bezier curve has all of the typical advantages of Bezier curves, such as efficient computation, subdivision, and variation diminution. We also discuss control of the speed of rotation, and cusp avoidance, both of which are simpler with our method.

This paper can be viewed as an extension of the classical work on interpolation of points (*i.e.*, position) to the interpolation of orientations.

Keywords: animation, orientation, interpolation, quaternion, Bezier curve.

1 Introduction

Smooth interpolation of three-dimensional object orientation, starting from n keyframe orientations, is used in computer animation to model moving solids, cameras, and lights. Shoemake clarified the superiority of unit quaternions as the representation of orientation in this setting [12], thus casting the problem as one of interpolation of n points on the quaternion sphere (the unit sphere in 4-space). Subsequently, many papers have been written solving the problem of constructing good interpolating curves on the quaternion sphere [12, 4, 8, 10, 11, 1], for orienta-

tion interpolation. However, all of these methods have constructed non-rational curves (using *slerp*ing, a trigonometric function, and/or constrained optimization). They have also lacked strong interactive control over the curve (*e.g.*, subdivision, local control, efficient redesign).

This paper shows how to construct a rational Bezier interpolating curve on the quaternion sphere, for orientation interpolation. Since this curve is a true Bezier spline (not an imitation of a Bezier curve as in Shoemake and others), it enjoys all of the advantages of Bezier curves, such as efficient computation, subdivision, local control, affine invariance, variation diminution, as well as a predictable behaviour and ease of implementation because of the rich understanding of Bezier curves. Since the curve has a complete analytic description, it allows simple manipulation and complete control. This construction answers many of the challenges for future work outlined by Shoemake in his paper.

Our method does not attempt to design the curve directly on the quaternion sphere as in other methods (which must apply restrictive constraints to stay on the sphere). Instead, the curve is initially designed freely in 4-space (using traditional interpolation techniques) and is then mapped to the sphere by a special rational map.

Related work is discussed in Section 2. Section 3 reviews the theory of quaternions. Sections 4-9 are the heart of the paper: Section 4 presents an outline of the new method, the map onto the quaternion sphere is developed in Section 5, its inverse in Section 6, and the map of a single cubic Bezier segment onto the sphere in Section 7. Ways to control the speed of rotation are presented in Section 8, and Section 9 discusses cusps. Examples of curves and animations generated by the method are presented in Section 10, and we close with some final thoughts in Section 11.

*This work was supported in part by the National Science Foundation under grant CCR-9213918.

2 Related work

Rather than discussing the approach of each of the other papers on orientation interpolation through curves on the quaternion sphere, it is enough to discuss a common tool of the methods: *slerping*. *Slerping* refers to spherical linear interpolation [12]: $\text{Slerp}(q_1, q_2; u) := \frac{\sin(1-u)\theta}{\sin\theta} q_1 + \frac{\sin u\theta}{\sin\theta} q_2$, where q_1 and q_2 are unit quaternions and θ is the angle between these two vectors. This achieves interpolation along a great arc of the quaternion sphere. It is clearly a non-rational, trigonometric map. Various papers have used various spline techniques based around replacement of linear interpolation by *slerping*: Bezier curve (Shoemake [12]), B-spline (Duff [4]), cardinal spline (Pletinckx [10]), Catmull-Rom spline (Schlag [11]).

The paper of Barr *et al.* [1] uses a different technique: constrained optimization to minimize tangential acceleration of the spherical curve. (It also uses *slerping* for interpolation.) It is also notable for its excellent motivation of the design of splines on non-Euclidean curved manifolds.

Our paper is strongly motivated by a paper of Dietz, Hoschek, and Jüttler [3] on the construction of interpolating curves on quadrics (including the sphere). Like the present paper, Dietz *et al.* map points from the sphere to 3-space, find an interpolating curve in 3-space, and map this curve back to the sphere. The major differences arise from the differences between 3-space and 4-space, and our particular attention to the use of the curves in animation (which lead to our analysis of cusps and speed control). Their map onto the sphere in 3-space is quite different than our map onto the sphere in 4-space. Also, we look at a single point of the map's inverse image rather than the entire line, which allows classical point interpolation methods to be applied in 4-space, rather than Dietz *et al.*'s system of equations approach to the interpolation of a curve through lines in 3-space.

3 Quaternions and the quaternion sphere

The theory of quaternions is well documented, such as in Shoemake [12] which also contains an excellent motivation of their advantages for representation of orientation. The relevant facts about quaternions for this paper are as follows. A quaternion is a 4-vector $(x_1, x_2, x_3, x_4) = x_1 + x_2 * i + x_3 * j + x_4 * k$, a 4-

dimensional analogue of complex numbers,¹ invented by Hamilton. A unit quaternion

$$(x_1, x_2, x_3, x_4) = \left(\cos \frac{\theta}{2}, v \sin \frac{\theta}{2}\right), \quad \|v\| = 1$$

corresponds to a rotation of θ about the axis v .²

Since a single rotation about an axis is sufficient to represent an arbitrary orientation of a solid object, unit quaternions are a representation for rigid body orientation. The other primary choices are the rotation matrix and Euler angles. Quaternions are the most elegant representation, at least for animation. Unlike Euler angles, quaternions have a unique representation for each orientation, do not experience gimbal lock, and can be combined easily. Unlike both Euler angles and rotation matrices, the quaternion has a concise representation (4 numbers) with a natural geometric analogue (through identification of the set of unit quaternions with the unit sphere S^3 in 4-space) which is highly useful for interpolation.

A major advantage of the unit quaternion is that we can control and predict the speed of rotation of the tumbling body, since the metrics of the sphere S^3 and rotation (the angular metric of $\text{SO}(3)$) are equivalent. That is, distance on the sphere is speed of rotation (*e.g.*, a constant speed path on the sphere yields a constant speed rotation of the object). We will explore this control in Section 8.

In the rest of the paper, a unit quaternion will be identified with a point on S^3 , the unit sphere in 4-space, which will be called the *quaternion sphere*.

4 Our method of orientation interpolation

We now present an outline of our method. The major idea is to design the interpolating curve freely in 4-space using traditional techniques and then map back onto the sphere (using a map M from 4-space onto the sphere). The input is a set of n orientations of a solid represented as unit quaternions q_1, \dots, q_n (Figure 1).

¹ i, j , and k each act very much like the imaginary number i : $i^2 = j^2 = k^2 = ijk = -1$.

²Two quaternions $[s_1, v_1]$ and $[s_2, v_2]$ (where v_i are 3-vectors) are multiplied by the formula: $[s_1, v_1] * [s_2, v_2] = [(s_1 * s_2 - v_1 \cdot v_2, s_1 * v_2 + s_2 * v_1 + v_1 \times v_2)]$. Representing a vector w as the quaternion $[0, w]$, the result of rotating w by the quaternion $q = [s, v]$ is $q^{-1}[0, w]q$, where $q^{-1} = ([s, -v]) / (s^2 + v \cdot v)$. This is the same result as rotating w about v by an angle $2\cos^{-1}(s)$.

1. If necessary, translate the orientations to unit quaternions.
2. (**Map quaternions into 4-space**) Map the quaternions q_i by M^{-1} into 4-space.
3. (**Interpolate in 4-space**) Interpolate the points $M^{-1}(q_i)$ in 4-space by a polynomial curve $C(t)$, for example a cubic B-spline.
4. (**Translate to Bezier spline**) Translate $C(t)$ to the equivalent cubic Bezier spline $C_{\text{bez}}(t)$.
5. (**Map back onto the sphere**) Map $C_{\text{bez}}(t)$ onto the sphere using M , one Bezier segment at a time, yielding a Bezier spline $D_{\text{bez}}(t)$.

$D_{\text{bez}}(t)$ is the desired spherical curve, a Bezier spline that interpolates the quaternions q_i . In our examples, since we use a cubic B-spline in step (3), $D_{\text{bez}}(t)$ is C_2 -continuous. Traditional techniques are used for steps (3) and (4), the interpolation in 4-space and the translation from polynomial curve to Bezier spline (see Farin [5]). Step 1 is also well understood (see Shoemake [12]). Steps 2 and 5 are the only steps that require elaboration. Step 2 is discussed in Section 6 and step 5 in Section 7.

Notice that any method of interpolation can be used in step (3), as long as it can be translated to a Bezier spline. We need to translate to a Bezier spline so that the method of step (5) can be applied.

5 Onto the sphere

We want a rational³ map from 4-space onto the unit sphere S^3 in 4-space. The challenge is to make the map rational, since it is simple to find non-rational maps onto the sphere (*e.g.*, normalization). A formula from number theory yields a solution [2][p. 318].

Lemma 5.1 (Euler, Aida)

$$\begin{aligned} (a^2 + b^2 + c^2 - d^2)^2 + (2ad)^2 + (2bd)^2 + (2cd)^2 \\ = (a^2 + b^2 + c^2 + d^2)^2 \end{aligned} \quad (1)$$

Notation 5.1

It is most natural to express the following map in projective 4-space, P^4 . $(x_1, x_2, x_3, x_4, x_5)$ in projective 4-space is equivalent to $(\frac{x_1}{x_5}, \frac{x_2}{x_5}, \frac{x_3}{x_5}, \frac{x_4}{x_5})$ in affine 4-space.

³A map $\alpha(t) = (x_1(t), \dots, x_k(t))$ is rational if each component $x_i(t)$ can be expressed as the quotient of two polynomials.

Corollary 5.1 *The map $M : P^4 \rightarrow S^3 \subset P^4$:*

$$M(x_1, x_2, x_3, x_4, 1) = \begin{pmatrix} x_1^2 + x_2^2 + x_3^2 - x_4^2 \\ 2x_1x_4 \\ 2x_2x_4 \\ 2x_3x_4 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 \end{pmatrix} \quad (2)$$

sends any point in projective 4-space onto the unit sphere in projective 4-space.⁴ That is, $\|M(x_1, x_2, x_3, x_4, 1)\| = 1$.

Remark 5.1 *To get a map from 4-space to the unit sphere, we are looking for a formula of the form $A^2 + B^2 + C^2 + D^2 = E^2$ where A, B, C, D and E are functions of at most four variables and at least one is a function of exactly four variables. Then a point in 4-space can be mapped to (A, B, C, D, E) where $\|(A, B, C, D, E)\| = 1$.*

The analogous map in 3-space is Lebesgue's $\mathcal{M}(x_1, x_2, x_3, x_4) = (2(x_1x_2 - x_3x_4), 2(x_2x_4 - x_1x_3), x_2^2 + x_3^2 - x_1^2 - x_4^2, \pm(x_1^2 + x_2^2 + x_3^2 + x_4^2))$ [2, 3], which is used by Dietz *et al.* [3].⁵ This is an elegant map since it is necessary and sufficient: every rational curve $c(t)$ on the unit sphere in 3-space has the form $\mathcal{M}(x_1(t), x_2(t), x_3(t), x_4(t))$, for some choice of $x_1(t), x_2(t), x_3(t), x_4(t)$. We are not as lucky in 4-space with our map M . But the fact that M does not map onto the set of all rational interpolating curves is not important: we need only insure that it creates a good interpolating curve for animation.

6 Back to 4-space

The inverse map M^{-1} is needed to map the quaternions q_i to $M^{-1}(q_i)$ as input to the interpolation in 4-space. If we find a curve interpolating $M^{-1}(q_i)$, then its image under M will be a curve interpolating q_i .

⁴The observant reader may notice that the image of the origin under M is undefined (and is the only such point). This is not a problem since we will see that the inverse images M^{-1} of unit quaternions all lie above the plane $x_4 = 0$ (with the sole exception of the identity quaternion $(1, 0, 0, 0, 1)$, which will map onto this plane but far from the origin) and the entire space curve interpolating the $M^{-1}(q)$ avoids the origin.

⁵These maps onto the sphere are instances of Pythagorean n -tuples. An interesting work on the use of Pythagorean triples for curve design is the work of Farouki and Sakkalis on Pythagorean hodograph curves [6].

Lemma 6.1 The map $M^{-1} : S^3 \rightarrow P^4$ is defined by

$$M^{-1}(X) = M^{-1}(x_1, x_2, x_3, x_4, x_5) = \quad (3)$$

$$\begin{cases} (x_2, x_3, x_4, x_5 - x_1, +2\sqrt{\frac{x_5-x_1}{2}}) & \text{if } X \neq (1, 0, 0, 0, 1) \\ & \text{(viz., if } x_1 \neq x_5) \\ \text{the hyperplane } x_4 = 0 & \text{otherwise}^6 \end{cases}$$

where $(x_1, x_2, x_3, x_4, x_5)$ lies on the unit sphere S^3 and only the positive square root is used.

Proof: If $x_1 \neq x_5$, $M^{-1}(M(p, q, r, s, 1)) = M^{-1}(p^2+q^2+r^2-s^2, 2ps, 2qs, 2rs, p^2+q^2+r^2+s^2) = (2ps, 2qs, 2rs, 2s^2, 2s) = (p, q, r, s, 1)$.

Since $M(x_1, x_2, x_3, 0, x_5) = (1, 0, 0, 0, 1)$, the inverse image of $(1, 0, 0, 0, 1)$ is the entire hyperplane $x_4 = 0$. ■

Corollary 6.1 M is surjective.

Proof: Every point of the unit sphere S^3 has an inverse image. ■

Note that $(1, 0, 0, 0, 1)$ is the identity quaternion, representing rotation by 0 degrees about an arbitrary axis.

When the quaternion $q_i = (q_{i1}, q_{i2}, q_{i3}, q_{i4})$ is mapped off of the sphere using M^{-1} , we apply the map $M^{-1}(q_{i1}, q_{i2}, q_{i3}, q_{i4}, 1)$. That is, we simply set the homogeneous coordinate to 1. This maps the quaternion to a unique point. The inverse of the projective point $\{(kq_{i1}, kq_{i2}, kq_{i3}, kq_{i4}, k)\}_{k \in \mathbb{R}}$ is a line,⁷ but we choose a unique preimage in order to apply point interpolation.

7 The image of a cubic Bezier segment

To take advantage of the Bezier representation, the Bezier spline in 4-space must be mapped back to the sphere as a Bezier curve, not simply a rational curve. It turns out that the image of each cubic Bezier segment of the spline is a sextic Bezier segment. The following lemma reveals the Bezier structure of these sextic image segments on the sphere. Notice that the structure of the map M is preserved in the control points (compare (4) and (2)).

⁶We arbitrarily choose $M^{-1}(1, 0, 0, 0, 1) = (1, 1, 1, 0, 1)$ in our implementation.

⁷ $M^{-1}(kq_{i1}, kq_{i2}, kq_{i3}, kq_{i4}, k) = (kq_{i2}, kq_{i3}, kq_{i4}, k(1 - q_{i1}), 2\sqrt{\frac{1-q_{i1}}{2}}\sqrt{k})$, or in affine space $\frac{\sqrt{k}}{\sqrt{2(1-q_{i1})}}(q_{i2}, q_{i3}, q_{i4}, 1 - q_{i1})$, so $M^{-1}\{(kq_{i1}, kq_{i2}, kq_{i3}, kq_{i4}, k)\}_{k \in \mathbb{R}}$ is the line from the origin through $(q_{i2}, q_{i3}, q_{i4}, 1 - q_{i1})$ in affine space.

Lemma 7.1 The image of a polynomial cubic Bezier segment $c(t)$ in 4-space under the map M is a rational Bezier segment of degree 6 with control points c_k and weights w_k ($k = 0, \dots, 6$):

$$c_k = \sum_{\substack{0 \leq i \leq 3 \\ 0 \leq j \leq 3 \\ i+j=k}} \frac{\binom{3}{i} * \binom{3}{j}}{\binom{6}{k}} c_{ijk} \quad (4)$$

$$c_{ijk} = \begin{pmatrix} (b_{i1}b_{j1} + b_{i2}b_{j2} + b_{i3}b_{j3} - b_{i4}b_{j4})/w_k \\ 2b_{i1}b_{j4}/w_k \\ 2b_{i2}b_{j4}/w_k \\ 2b_{i3}b_{j4}/w_k \end{pmatrix}$$

$$w_k = \sum_{\substack{0 \leq i \leq 3 \\ 0 \leq j \leq 3 \\ i+j=k}} \frac{\binom{3}{i} * \binom{3}{j}}{\binom{6}{k}} w_{ijk}$$

$$w_{ijk} = (b_{i1}b_{j1} + b_{i2}b_{j2} + b_{i3}b_{j3} + b_{i4}b_{j4})$$

where $b_i = (b_{i1}, b_{i2}, b_{i3}, b_{i4})$ are the control points of $c(t)$ ($i = 0, 1, 2, 3$).

Proof: Let $M(c(t)) = M(\sum_{i=0}^3 B_i^3(t)b_i) := (m_1(t), m_2(t), m_3(t), m_4(t), m_5(t))$ (where $B_i^n(t)$ is the i^{th} Bernstein polynomial of degree n). Consider the coordinate

$$\begin{aligned} m_5(t) &= [\sum_{i=0}^3 B_i^3(t)b_{i1}]^2 + \dots + [\sum_{i=0}^3 B_i^3(t)b_{i4}]^2 \\ &= \sum_{i=0}^3 \sum_{j=0}^3 \frac{\binom{3}{i} * \binom{3}{j}}{\binom{6}{i+j}} B_{i+j}^6(t) (b_{i1}b_{j1} + \dots + b_{i4}b_{j4}) \end{aligned}$$

by the product rule of Bernstein polynomials [5]. Letting $k = i + j$, this becomes

$$\sum_{k=0}^6 B_k^6(t) \sum_{\substack{0 \leq i \leq 3 \\ 0 \leq j \leq 3 \\ i+j=k}} \frac{\binom{3}{i} * \binom{3}{j}}{\binom{6}{k}} (b_{i1}b_{j1} + \dots + b_{i4}b_{j4})$$

Computing the other coordinates analogously yields

$$M(c(t)) = \sum_{k=0}^6 B_k^6(t) \sum_{\substack{0 \leq i \leq 3 \\ 0 \leq j \leq 3 \\ i+j=k}} \frac{\binom{3}{i} * \binom{3}{j}}{\binom{6}{k}} \begin{pmatrix} b_{i1}b_{j1} + b_{i2}b_{j2} + b_{i3}b_{j3} - b_{i4}b_{j4} \\ 2b_{i1}b_{j4} \\ 2b_{i2}b_{j4} \\ 2b_{i3}b_{j4} \\ b_{i1}b_{j1} + b_{i2}b_{j2} + b_{i3}b_{j3} + b_{i4}b_{j4} \end{pmatrix}$$

which is a sextic rational Bezier curve with control points (4) and weights w_k . ■

Figure 2 shows the control polygon of the rational Bezier curve on the sphere. Notice that the control polygon does not in general lie on the sphere, only the associated Bezier curve.

8 Control of speed

To control the speed of rotation of the tumbling body, we would like to control our speed along the spherical curve (see Section 3). With our rational Bezier spherical curve, we cannot solve Shoemake’s open challenge of designing a spherical curve parameterized by arc length [12], which would generate perfectly regular changes of orientation, since no rational curve can be parameterized by a rational function of its arc length (Farouki and Sakkalis [7]). However, we have a great deal of control over our speed on the spherical curve: through the knot sequence. Moreover, this control is very simple and intuitive.

8.1 Choice of knot sequence

Since the spherical curve that we create using Lemma 7.1 is a Bezier spline, we can use its knot sequence to control the speed along the curve. For example, we might like to approximate an arc-length parameterization (since it is impossible to attain exactly) for a uniform change of orientation. Chord-length parameterizations assign knot intervals proportional to the Euclidean distance between data points. Since our points and curve lie on a sphere, it is more appropriate to use a non-Euclidean metric measuring distance on the sphere:

$$\text{dist}(A, B) = \theta r = \theta = \cos^{-1}(A \cdot B)$$

(where θ is the angle subtended by the points A and B on the unit sphere). Non-Euclidean variants of other parameterizations, such as centripetal, could also be used.

In many cases, perfectly regular tumbling will not be our goal, and manipulation of the knot sequence can just as easily achieve other effects. The above non-Euclidean metric will still be useful in these contexts.

This control of the speed along the curve via the knot sequence is another benefit of using a Bezier representation of the spherical curve.

8.2 Choice of frames based on arc length

Another simple mechanism for effective speed control in the animation is to choose intermediate frames based on arc-length. Suppose that, for example, we wish to simulate constant speed along the orientation curve (*i.e.*, arc length parameterization). Based on computation of arc length, the orientation of intermediate frames can be chosen at approximately equal spacing along the orientation curve. The length of a segment should be computed by subdivision (until an acceptable degree of accuracy is achieved) rather than by exact computation via the integral $\int_a^b \|\mathcal{C}'(t)\| dt$ (for the segment $\mathcal{C}(t)$). This method is possible since we have a closed form Bezier representation of the spherical curve. Other methods, which do not have direct representation of the entire curve cannot perform this type of arc-length based choice of intermediate frames.

9 Cusps

A cusp in a spherical curve is associated with an abrupt change of orientation, undesirable in an animation. All of the spherical curves created by other methods can contain cusps (or what Shoemake calls ‘kinks’), and so can ours. However, the next lemma shows that our spherical curves rarely have cusps. Since it is simple to design our original 4-space curve without cusps (a nonplanar cubic Bezier curve cannot contain a cusp [5]), the only cusps are those introduced by the map M . The following lemma shows that M only introduces cusps in unusual situations, which should be easy to avoid during design (especially since design is interactive with our method).

Lemma 9.1 *M introduces a cusp in the curve $C(t)$ at $t = t_0$ if one of the following conditions holds:*

- $C(t_0) = (0, 0, 0, 0)$
- $C'(t_0) = (0, 0, 0, 0)$
- $C(t_0) \cdot C'(t_0) = 0$ and $C_4(t_0) = C'_4(t_0) = 0$
- $\frac{\partial}{\partial t}(M_1(C(t_0))) = kM_1(C(t_0))$, where M_1 is defined below and $k = \frac{\frac{\partial}{\partial t}(M_1(C)) \cdot M_1(C)}{M_1(C) \cdot M_1(C)}$.

Proof: M can be expressed as the composition of two maps (in affine space):

$$M_1 : (p, q, r, s) \rightarrow (p^2 + q^2 + r^2 - s^2, 2ps, 2qs, 2rs)$$

and

$$M_2 : V \rightarrow V/\|V\|$$

The problem now reduces to determining when these two maps introduce cusps. We will show that $M_1(C(t_0))$ is a cusp when $C(t_0)$ or $C'(t_0)$ is the origin or $C(t_0) \cdot C'(t_0) = 0$; while $M_2(C(t_0))$ is a cusp when $C'(t_0) = kC(t_0)$, $k \in \mathfrak{R}$.

Consider the map M_1 . A curve $C(t)$ has a cusp at $t = t_0$ if $\frac{\partial}{\partial t}(C(t_0)) = (0, 0, 0, 0)$. Let $C(t) = (p(t), q(t), r(t), s(t))$. Suppose that $M_1(C(t_0))$ is a cusp.

$$\begin{aligned} \frac{\partial}{\partial t}(M_1(p, q, r, s)) &= \begin{pmatrix} 2pp' + 2qq' + 2rr' - 2ss' \\ 2p's + 2ps' \\ 2q's + 2qs' \\ 2r's + 2rs' \end{pmatrix} \\ &= (0, 0, 0, 0) \end{aligned} \quad (5)$$

If $s \neq 0$, $p' = -p(s'/s)$, $q' = -q(s'/s)$, $r' = -r(s'/s)$, and substituting into $2pp' + 2qq' + 2rr' - 2ss' = 0$ yields $\frac{-2s'}{s}(p^2 + q^2 + r^2 + s^2) = 0$. If $s' \neq 0$, this reduces to $p^2 + q^2 + r^2 + s^2 = 0$ or (p, q, r, s) is the origin.

If $s = 0$ and $s' \neq 0$, (5) again reduces to $(p, q, r, s) = (0, 0, 0, 0)$. If $s = s' = 0$, (5) reduces to $pp' + qq' + rr' = 0$ (or $(p, q, r, s) \cdot (p', q', r', s') = 0$). If $s \neq 0$ and $s' = 0$, (5) reduces to $(p', q', r', s') = (0, 0, 0, 0)$.

Next consider M_2 . Suppose that $M_2(C(t_0))$ is a cusp. We assume that $C(t_0) \neq (0, 0, 0, 0)$.

$$\begin{aligned} \frac{\partial}{\partial t}(M_2(C(t_0))) &= \frac{\partial}{\partial t}\left(\frac{C(t_0)}{\|C(t_0)\|}\right) = \frac{\partial}{\partial t}\left(\frac{C(t_0)}{\sqrt{C(t_0) \cdot C(t_0)}}\right) \\ &= \frac{\|C\|C' - \left(\frac{C \cdot C'}{\sqrt{C \cdot C}}\right)C}{C \cdot C} = (0, 0, 0, 0) \end{aligned}$$

Multiplying by $\|C(t_0)\|^3$,

$$\begin{aligned} \|C\|^2 C' - (C \cdot C')C &= (0, 0, 0, 0) \\ C' &= kC \end{aligned}$$

where $k = \frac{C \cdot C'}{C \cdot C}$. In other words, the map M_2 only introduces cusps into the curve $C(t)$ when $C'(t_0) = kC(t_0)$.

Note that M_2 preserves cusps: that is, if $C(t_0)$ is a cusp, then $M_2(C(t_0))$ is also a cusp. ■

Thus, M introduces a cusp in $C(t)$ when the curve $C(t)$ passes through the origin, or its hodograph $C'(t)$ passes through the origin. M also introduces a cusp if $C(t_0)$ and $C'(t_0)$, the vectors to the curve and hodograph *at the same parameter value* t_0 , are orthogonal as well as lie in the hyperplane $x_4 = 0$. Finally, a cusp can be introduced if the vectors to the

curve $M_1(C(t))$ and its tangent, at the same parameter value t_0 , are multiples in the special ratio k of the theorem. These are all pathological occurrences which most curves will not contain, and moreover they are unstable conditions which can be removed by manipulation of the original 4-space curve.

10 Examples

The Bezier nature of our spherical curves predicts good quality curves (*e.g.*, variation-diminishing). Moreover, our spherical curve will be C^2 continuous, since it is the image under a rational map of a C^2 -continuous cubic B-spline. This quality is supported by our practical experience. The curves that we have generated are well-behaved.

We present an example of a tumbling maple leaf. Figure 1 shows the input to our animation problem: n orientations of n keyframes of a solid. The orientations are shown on the left as red unit quaternions on the quaternion sphere, with the associated keyframes on the right. Figure 2 shows the interpolating rational Bezier curve on the quaternion sphere as determined by our method (on the left) and the animation corresponding to this spherical curve (on the right). In this static picture, we only show a few of the intermediate frames generated by the spherical curve. The control polygon of the Bezier curve is also drawn, in black. We visualize the curves in 4-space by using quaternions with $x_3 = 0$, thus allowing projection onto the 3-dimensional hyperplane $x_3 = 0$.⁸ This is purely for purposes of visualization: all computations are in 4-space.

The construction of the orientation curve is illustrated in Figure 3. The input quaternions are drawn in red. They are mapped by M^{-1} to the blue points, which are interpolated freely in 4-space by the blue Bezier curve (with its control polygon). Finally, the blue space curve is mapped back onto the sphere by M to the red spherical curve, which interpolates the original quaternions.

Notice that the spherical curve only controls the orientation of the frames. The position of the tumbling leaf in each frame is controlled by a second interpolating curve. It is impossible to visualize the change in orientation unless the object also moves through 3-space.

⁸Quaternions with $x_3 = 0$ are mapped by M^{-1} to points in 4-space with $x_2 = 0$, so the entire interpolation in 4-space lies in the plane $x_2 = 0$ and maps back to a spherical curve entirely in the hyperplane $x_3 = 0$.

Finally, we remind the reader that the representation of an orientation by a quaternion is not unique: antipodal quaternions represent the same orientation. In constructing the quaternion representation of the series of keyframe orientations, it is wise to use this degree of freedom and choose whichever quaternion lies closest (on the sphere) to the previous quaternion, so that the animation does not perform undesired flips. For example, if two consecutive identical orientations are represented by two antipodal quaternions, the object will perform a full rotation rather than remaining stationary.

11 Conclusions

We have developed a way of controlling orientation rationally, by developing rational Bezier curves that interpolate quaternion orientations. Control of orientation by a rational Bezier spline is more efficient and more amenable to manipulation (*e.g.*, alteration of the curve, control of speed) than control by a non-rational curve for which no direct, closed-form representation is known.

By constructing a rational map from 4-space to the quaternion sphere, and the inverse map from the sphere to 4-space, we reduce orientation interpolation on the sphere to point interpolation in 4-space. The Bezier structure of the curve in 4-space is mapped to the sphere. Point interpolation in 4-space can be performed using any classical method. We have chosen a cubic B-spline, since it is perhaps the most widely understood method, but other methods could certainly be used. Since it applies traditional techniques of point interpolation, our method can be viewed as an extension of the interpolation of position (in 3-space) to the interpolation of orientation (in 4-space).

12 Acknowledgements

We thank the anonymous referees for their comments.

References

- [1] Barr, A.H., B. Currin, S. Gabriel and J.F. Hughes (1992) Smooth interpolation of orientations with angular velocity constraints using quaternions. SIGGRAPH '92, Chicago, 26(2), 313–320.
- [2] Dickson, L.E. (1952) History of the theory of numbers: Volume II, Diophantine analysis. Chelsea (New York).
- [3] Dietz, R., J. Hoschek and B. Jüttler (1993) An algebraic approach to curves and surfaces on the sphere and on other quadrics. Computer Aided Geometric Design 10, 211–229.
- [4] Duff, T. (1985) Quaternion splines for animating orientation. Proceedings of the Monterey Computer Graphics Workshop, 54–62.
- [5] Farin, G. (1993) Curves and surfaces for computer aided geometric design. Academic Press (New York), third edition.
- [6] Farouki, R.T. and T. Sakkalis (1990) Pythagorean hodographs. IBM Journal of Research and Development 34, 736–752.
- [7] Farouki, R.T. and T. Sakkalis (1991) Real rational curves are not "unit speed". Computer Aided Geometric Design 8, 151–157.
- [8] Gabriel, S. and J. Kajiya (1985) Spline interpolation in curved space. In SIGGRAPH '85 course notes 'State of the art image synthesis'.
- [9] Misner, C.W., K.S. Thorne and J.A. Wheeler (1973) Gravitation. W.H. Freeman (San Francisco).
- [10] Pletinckx, D. (1989) Quaternion calculus as a basic tool in computer graphics. The Visual Computer 5, 2–13.
- [11] Schlag, J. (1991) Using geometric constructions to interpolate orientation with quaternions. In Graphics Gems II, Academic Press (New York), 377–380.
- [12] Shoemake, K. (1985) Animating rotation with quaternion curves. SIGGRAPH '85, San Francisco, 19(3), 245–254.

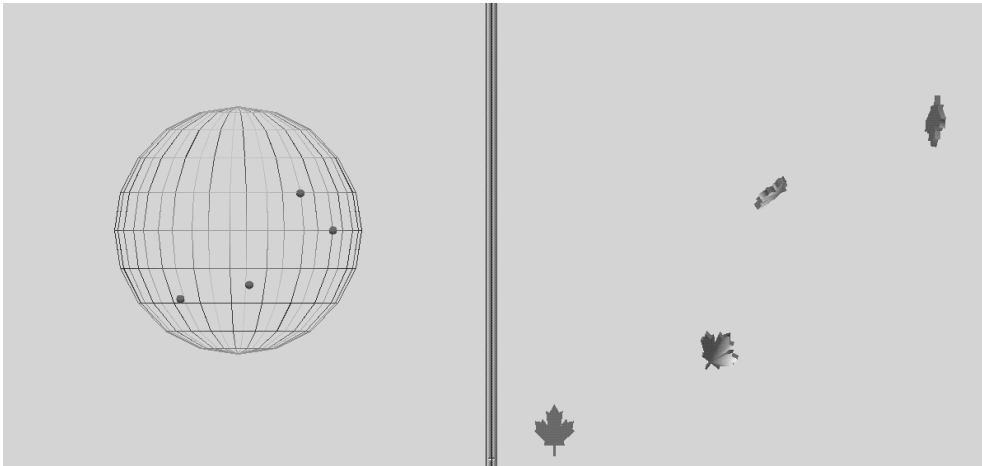


Figure 1: Input to the animation problem

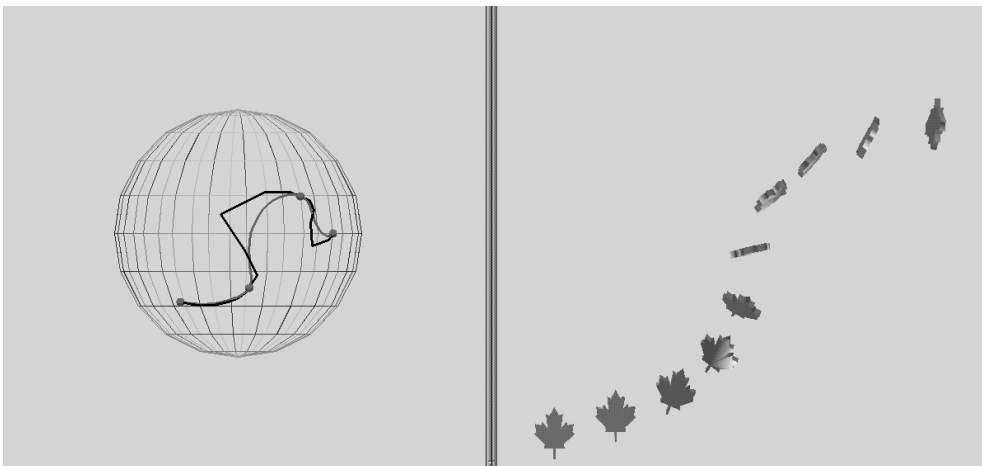


Figure 2: Orientation curve and animation

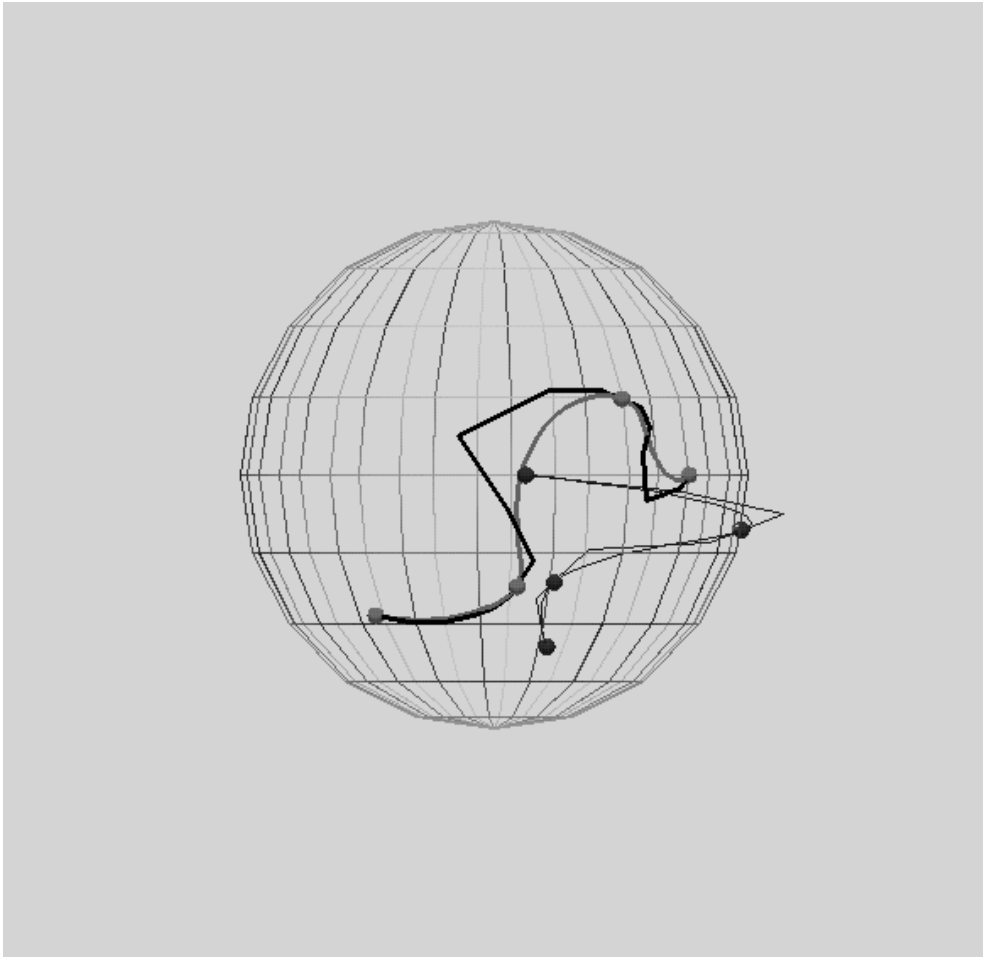


Figure 3: Mapping on and off the quaternion sphere, using M