A Geometrically Based Approach to 3D Skeleton Curve Blending

Francis Lazarus

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

A Geometrically Based Approach to 3D Skeleton Curve Blending

Francis Lazarus

N° 2458

Janvier 1995

PROGRAMME 4

Robotique,

image

et vision

ISSN 0249-6399

A Geometrically Based Approach to 3D Skeleton Curve Blending

Francis Lazarus*

Programme 4 — Robotique, image et vision
Projet Syntim

Rapport de recherche n° 2458 — Janvier 1995 — 20 pages

Abstract: This paper presents an efficient method to smoothly transform one polyline into another. The method makes use of a moving adapted frame associated to each curve. We introduce a simple propagation equation involving the moving frame which accounts for the transition between two consecutive points of a polyline. The curve transformation is performed using the interpolation of the quantities involved in this propagation equation. Depending on the type of interpolation, we propose two algorithms to compute the intermediary shape of the curve. Moreover, we define a measure of the transformation and minimize this measure to distinguish a "minimal transformation" which the user can modify through the addition of simple parameters.

Key-words: Curve, Computer Animation, Computer-Aided Geometric Design, Interpolation, Deformation, Shape transformation.

(Résumé : tsvp)

*Email : Francis.Lazarus@inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Une approche géométrique pour l'animation de courbes polygonales gauches

Résumé : Nous présentons une approche originale pour calculer une transformation (homotopie) entre deux courbes polygonales gauches. Pour cela nous munissons chaque courbe d'un repère mobile. Nous introduisons ensuite une équation de propagation simple qui explicite le passage entre deux sommets consécutifs d'une courbe et dans laquelle intervient le repère mobile. La transformation est alors effectuée à l'aide de l'interpolation des quantités apparaissant dans cette équation et nous obtenons deux algorithmes en fonction du type d'interpolation choisi pour le repère mobile. De plus, nous définissons une mesure de la transformation que nous minimisons afin de définir une "transformation minimale", modifiable à l'aide de paramètres simples.

Mots-clé : Courbes, Animation, CAO, Interpolation, Déformation, Modélisation géométrique

1 Introduction

The animation and deformation of closed plane curves have been widely studied in image analysis [KWT87, KTZ92, ST94] and in computer graphics [SGWM93, SG92]. However, it seems that little attention has been paid in the computer graphics community, to the problem of the transformation of 3D open curves, even though there are many applications for this in animation techniques. For instance, generalized cylinders or implicit surfaces defined by skeleton curves, may be animated through the use of their skeleton [BW76, Wyv90]. We can also animate the axial deformations proposed in [LCJ94] or use the curve transformations to perform a metamorphosis as in [LV94].

This paper presents an original approach to analyze curve transformations and proposes two simple algorithms that, given two polylines with n vertices, compute an intermediary polyline.

Our transformation techniques has several advantages:

- The curve transformation is very smooth.
- The interpolation of the geometric entities such as length, curvature or torsion is well-controlled.
- The geometry of the interpolated curve is independent of the relative position of the two initial curves in space.
- The computation is fast enough to allow the visualization of the curve animation in fully interactive time.

In this paper, we will assume that the polylines have been obtained by regularly sampling smooth regular parametrized curves. Note that this assumption is not restrictive, since it is always possible to fit a polyline to an interpolating spline. One can also assume that instead of two polylines, the user has been given two parametrized curves with which he or she wants to get a continuous transformation of the curves.

As the motion of a curve in the 3D space may be more complex than in the plane, we provide each curve with an associated moving frame, so that we transform both the curves and their frames. In section 2 we show, in a theoretical framework, that the curves may be represented by a progression equation involving their moving frame and we propose two methods to interpolate these frames.

As the addition of a moving frame may seem unnatural we point out in section 3, the relationship between the geometry of the curve and its frame, when a particular frame is chosen.

In section 4, we propose a measure of the deformation and deduce from this, by a minimization process, a simple way to interpolate the moving frames. Section 5 presents extensions to modify this "minimal interpolation" while section 6 presents our implementation and the results.

2 Our approach

A classic way to study the geometry of a curve is to define an *adapted* frame along the curve, whose members are either tangent to or perpendicular to the curve [Bis75, Fau93].

As mentionned in the introduction, we consider a parametrized curve Γ, defined on the interval [0,1], sampled with the step $\frac{1}{n}$. If $\mathcal{R}$ is an adapted frame for Γ, we can express $\Gamma(\frac{i+1}{n})$, $(i = 0, 1, ..., n-1)$, in the local frame $(\Gamma(\frac{i}{n}), \mathcal{R}(\frac{i}{n}))$ by the following progression equation (see Figure 1)

$$\Gamma(\frac{i+1}{n}) = \Gamma(\frac{i}{n}) + \frac{1}{n}\mathcal{R}(\frac{i}{n})\gamma(\frac{i}{n}) \tag{1}$$

Now, given two curves Γ_0 and Γ_1 we can construct a curve Γ_t, going from Γ_0 to Γ_1. To do

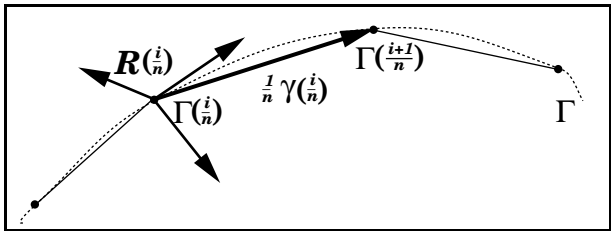

Figure 1: From $\Gamma(\frac{i}{n})$ to $\Gamma(\frac{i+1}{n})$.

so we interpolate the quantities $(\gamma_t(\frac{i}{n}), \mathcal{R}_t(\frac{i}{n}))$ from $(\gamma_0(\frac{i}{n}), \mathcal{R}_0(\frac{i}{n}))$ to $(\gamma_1(\frac{i}{n}), \mathcal{R}_1(\frac{i}{n}))$, and then recover the polyline $\Gamma_t(\frac{i}{n})$ at any time t in [0,1], using equation 1.

In section 2.1 we will show that we can interpolate separately $\gamma_t(\frac{i}{n})$ and $\mathcal{R}_t(\frac{i}{n})$ and that this process is relevant when the sampling step $\frac{1}{n}$ tends to 0. In the following two sections 2.2 and 2.3 we will propose two different ways to interpolate the frame $\mathcal{R}_t(\frac{i}{n})$.

2.1 A convergent process

As we desire a smooth transformation, we must be sure when the sampling step $\frac{1}{n}$ tends to 0, that the reconstructed curve $\Gamma_t(\frac{i}{n})$ tends to a differentiable curve.

In order to establish this result we need some definitions. A sequence f^n of functions, defined on [0,1], is said to *converge uniformly* to f if

$$\lim_{n\to\infty} \sup_{s\in[0,1]} \|f(s) - f^n(s)\| = 0$$

Furthermore, if $(f_i^n)_{0\leq i\leq n}$ is a sequence of sequences, with values in some $\mathbb{R}^p$, we will say that f_i^n *converges uniformly* to f if the sequence of functions $s \mapsto f^n_{\lfloor sn \rfloor}$ converges uniformly

to f.

Let Γ_0 and Γ_1 be two regular parametrized curves and $\mathcal{R}_0$ and $\mathcal{R}_1$ their respective frames, we put (see Figure 2)

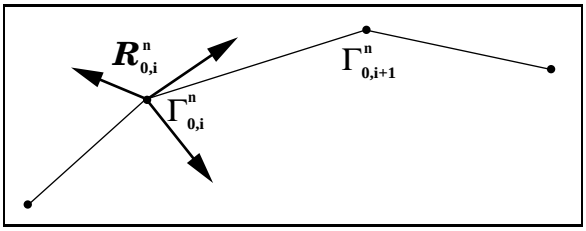

Figure 2: The sampled curve Γ_0

$$\Gamma^n_{0,i} = \Gamma_0(\frac{i}{n}) \qquad \text{and} \qquad \Gamma^n_{1,i} = \Gamma_1(\frac{i}{n}) \ , \ i \in [0,n]$$
$$\mathcal{R}^n_{0,i} = \mathcal{R}_0(\frac{i}{n}) \qquad \text{and} \qquad \mathcal{R}^n_{1,i} = \mathcal{R}_1(\frac{i}{n}) \ , \ i \in [0,n]$$

We also define:

$$\Delta^n_{0,i} = n(\Gamma^n_{0,i+1} - \Gamma^n_{0,i}) \qquad \text{and} \qquad \Delta^n_{1,i} = n(\Gamma^n_{1,i+1} - \Gamma^n_{1,i}) \ , \ i \in [0,n-1]$$
$$\gamma^n_{0,i} = {\mathcal{R}^n_{0,i}}^{-1} \Delta^n_{0,i} \qquad \text{and} \qquad \gamma^n_{1,i} = {\mathcal{R}^n_{1,i}}^{-1} \Delta^n_{1,i} \ , \ i \in [0,n-1]$$

Since Γ_j $(j = 0,1)$ is defined on the compact set [0,1], we can stipulate that $\Gamma^n_{j,i}$, $\mathcal{R}^n_{j,i}$, $\Delta^n_{j,i}$, $\gamma^n_{j,i}$ converge uniformly to the respective functions Γ_j, $\mathcal{R}_j$, Γ'_j, $v_j\,\vec{\imath}$, where $v_j(s)$ is the velocity at $\Gamma_j(s)$[1] and $\vec{\imath}$ is the first vector of the canonical frame.

Now, let us suppose that we know how to interpolate $\gamma^n_{j,i}$ and $\mathcal{R}^n_{j,i}$ in the function of time t, so that we are given $\gamma^n_{t,i}$ and $\mathcal{R}^n_{t,i}$ such that

$$\gamma^n_{t=0,i} = \gamma^n_{0,i} \qquad \text{and} \qquad \gamma^n_{t=1,i} = \gamma^n_{1,i}$$
$$\mathcal{R}^n_{t=0,i} = \mathcal{R}^n_{0,i} \qquad \text{and} \qquad \mathcal{R}^n_{t=1,i} = \gamma^n_{1,i}$$

and such that $\gamma^n_{t,i}$ and $\mathcal{R}^n_{t,i}$ converge uniformly to differentiable functions $\gamma_t(s)$ and $\mathcal{R}_t(s)$. Hence we see that $\Delta^n_{t,i} = \mathcal{R}^n_{t,i}\,\gamma^n_{t,i}$ converges uniformly to $\Delta_t = \mathcal{R}_t\,\gamma_t$. Finelly we can define $\Gamma^n_{t,i}$ and Γ_t by

$$\Gamma^n_{t,0} = \Gamma_t(0), \ t \in [0,1]$$

[1]Since Γ_j is regular, v_j is strictly positive, i.e. $v_j = \|\Gamma'_j\|$.

$$\Gamma^n_{t,i+1} = \Gamma^n_{t,i} + \frac{\Delta^n_{t,i}}{n} = \Gamma^n_{t,0} + \frac{1}{n}\sum_{k=0}^{i} \Delta^n_{t,k}$$

and

$$\Gamma_t(s) = \Gamma_t(0) + \int_0^s \Delta_t(u)\, du$$

In Appendix A, we show that $\Gamma^n_{t,i}$ converges uniformly to Γ_t. Moreover we have $\Gamma_t^{'} = \Delta_t = \mathcal{R}_t\, \gamma_t$.

Hence, if $\gamma_t = v_t\, \vec{\imath}$, we conclude that the velocity of Γ_t is v_t and that $\mathcal{R}_t$ is an adapted frame at any time t.

Furthermore, when $(s,t) \mapsto \gamma_t(s)$ and $(s,t) \mapsto \mathcal{R}_t(s)$ are continuous , the mapping $(s,t) \mapsto \Gamma_t(s)$ defines an homotopy from the curve Γ_0 to the curve Γ_1.

2.2 An interpolation scheme for the moving frame

So far we have seen that we could separately interpolate the quantities $\gamma^n_{t,i}$ and $\mathcal{R}^n_{t,i}$. The 3D vector $\gamma^n_{t,i}$ may be obtained simply by a linear interpolation

$$\gamma^n_{t,i} = (1-t)\ \gamma^n_{0,i} + t\ \gamma^n_{1,i} \tag{2}$$

so that $\gamma^n_{t,i}$ converges uniformly to $((1-t)\, v_0(s) + t\, v_1(s))\ \vec{\imath}$, where $v_0(s)$ (resp. $v_1(s)$) is the velocity at $\Gamma_0(s)$ (resp. $\Gamma_1(s)$). From this we see that **the time evolution of the length is linear**

$$\mathcal{L}(\Gamma_t) = \int_0^1 \|\Gamma_t^{'}\| \ = \ \int_0^1 ((1-t)\, v_0 + t\, v_1) \ = \ (1-t)\, \mathcal{L}(\Gamma_0) + t\, \mathcal{L}(\Gamma_1)$$

It only remains to find an interpolation scheme for the frame $\mathcal{R}^n_{t,i}$, which may be equally considered as a rotation. A study of moving frames shows that for any differentiable path $\mathcal{R}(s)$ in the space SO_3 of rotations we have

$$\mathcal{R}^{'}(s) = \mathcal{R}(s)\, \tilde{\mathbf{\Omega}}(s)$$

where $\mathbf{\Omega}$ is[2] the angular velocity of $\mathcal{R}$. Note that we choose to multiply $\mathcal{R}(s)$ by $\tilde{\mathbf{\Omega}}(s)$ on the left, so that the components of $\mathbf{\Omega(s)}$ that appear in $\tilde{\mathbf{\Omega}}(s)$ should be expressed in the local frame $\mathcal{R}(s)$.

Hence in order to interpolate two paths $\mathcal{R}_0(s)$ and $\mathcal{R}_1(s)$ in SO_3 we need only to interpolate their angular velocity and to solve the ordinary differential equation (ODE)

$$\mathcal{R}_t^{'}(s) = \mathcal{R}_t(s)\, \tilde{\mathbf{\Omega}}_t(s) \tag{3}$$

where $\mathbf{\Omega}_t$ is an interpolation of $\mathbf{\Omega}_0$ and $\mathbf{\Omega}_1$ and where the initial condition $\mathcal{R}_t(0)$ is given by an interpolation of the two initial frames $\mathcal{R}_0(0)$ and $\mathcal{R}_1(0)$.

[2]From now on, we will denote $\tilde{\mathbf{v}}$ the map $\mathbf{u} \mapsto \tilde{\mathbf{v}}\mathbf{u} = \mathbf{v} \times \mathbf{u}$ where $\times$ denotes the cross product.

2.2.1 Solving[3] $\mathcal{R}' = \mathcal{R}\,\tilde{\mathbf{\Omega}}$

In general the numerical solution of an ODE is obtained by the Euler method [PFTV92] (or any derivative method like Runge-Kutta) which advances the solution from $\mathcal{R}_i$ to $\mathcal{R}_{i+1}$ by

$$\mathcal{R}_{i+1} = \mathcal{R}_i + \frac{1}{n}\,\mathcal{R}_i\,\tilde{\mathbf{\Omega}}(\frac{i}{n}) \tag{4}$$

Meanwhile, such a method does not guaranty that $\mathcal{R}_{i+1}$ is still a rotation. This arises from the fact that equation 3 is defined over the rotation space SO_3 (a manifold embedded in $\mathbb{R}^9$, homeomorphic to the projective space $\mathbb{P}_3$) and that the Euler method should only be used on an open set of a vector space. If we still wish to use the Euler method, we need to map into a parameter space, through the use of a *chart*.

Hence, if (V, ϕ) is a *chart* of SO_3 around a rotation $\mathcal{R}_0$ and $g = \phi^{-1}$ is the corresponding parametrization (see Figure 3), we should solve in the parameter space

$$\mathbf{v}' = (dg)^{-1}.g(\mathbf{v})\,\tilde{\mathbf{\Omega}} \tag{5}$$

with initial condition $\mathbf{v}_0 = \phi(\mathcal{R}_0)$.

Note that $(dg)^{-1}.g(\mathbf{v})\,\tilde{\mathbf{\Omega}}$ is valid, since $g(\mathbf{v})\,\tilde{\mathbf{\Omega}}$ lies in the tangent space of SO_3 at $g(\mathbf{v})$, which is isomorphic to the product of $g(\mathbf{v})$ by the space of antisymmetric matrices.

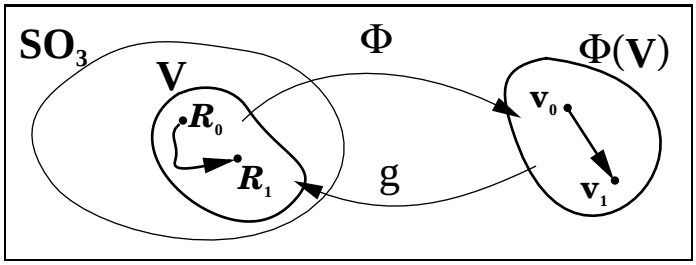

Figure 3: A chart over a neighborhood V of rotation $\mathcal{R}_0$

In order to parametrize SO_3 we use the exponential map defined by

$$\mathbf{v} \mapsto exp(\tilde{\mathbf{v}}) = Id + \tilde{\mathbf{v}} + \frac{\tilde{\mathbf{v}}^2}{2\,!} + ... = Id + \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\,\tilde{\mathbf{v}} + \frac{1-\cos\|\mathbf{v}\|}{\|\mathbf{v}\|^2}\,\tilde{\mathbf{v}}^2$$

where $\|.\|$ is the Euclidean norm. The remarkable character of this map may be seen by the conjunction of the following properties: its definition is algebraic, it is analytic, and it corresponds to the geometric definition of the exponential map that involves geodesics (see [Car76], p. 284).

[3] In this part we omit the parameter t and the superscript n to clarify the notations.

This parametrization is defined around the identity matrix but may be "shifted" around any matrix $\mathcal{R}$ with the parametrization

$$\mathbf{v} \mapsto \mathcal{R}\, exp(\tilde{\mathbf{v}})$$

The exponential map may be differentiated by using the formula [Fau89]

$$dexp(\tilde{\mathbf{v}}).\mathbf{h} = \mathcal{R}\,\tilde{\mathbf{b}}(\mathbf{v},\mathbf{h})$$

with

$$\mathbf{b}(\mathbf{v},\mathbf{h}) = \frac{\|\mathbf{v}\| - \sin\|\mathbf{v}\|}{\|\mathbf{v}\|^3} < \mathbf{v},\mathbf{h} > \mathbf{v} + \frac{\sin\|\mathbf{v}\|}{\|\mathbf{v}\|}\mathbf{h} + \frac{1-\cos\|\mathbf{v}\|}{\|\mathbf{v}\|^2}\mathbf{h}\times\mathbf{v}$$

where $<,>$ denotes the dot product. Identifying $\mathcal{R}\,\tilde{\mathbf{b}}(\mathbf{v},\mathbf{h})$ and $\mathcal{R}\,\tilde{\mathbf{\Omega}}$ we can extract $\mathbf{h}$

$$\mathbf{h} = (dexp(\tilde{\mathbf{v}}))^{-1}.\mathcal{R}\,\tilde{\mathbf{\Omega}} = \mathbf{a}(\mathbf{v},\mathbf{\Omega})$$

where

$$\mathbf{a}(\mathbf{v},\mathbf{h}) = \mathbf{h} + \frac{1}{2}\mathbf{v}\times\mathbf{h} + (\frac{1}{\|\mathbf{v}\|^2} - \frac{\sin\|\mathbf{v}\|}{2\|\mathbf{v}\|(1-\cos\|\mathbf{v}\|)})\mathbf{v}\times(\mathbf{v}\times\mathbf{h})$$

From equation 5, we now see that a "good" Euler method may be performed by the following scheme

$$\begin{cases} \mathbf{v}_0 = 0 \\ \mathcal{R}_i = \mathcal{R}(0)\, exp(\tilde{\mathbf{v}}_i) \\ \mathbf{v}_{i+1} = \mathbf{v}_i + \frac{1}{n}\,\mathbf{a}(\mathbf{v}_i,\mathbf{\Omega}_i) \end{cases}$$

Note that we have shifted the parametrization around the initial condition $\mathcal{R}(0)$. Meanwhile we can shift the parametrization around $\mathcal{R}_i$ at each step. Since $\mathbf{a}(\mathbf{0},\mathbf{\Omega}) = \mathbf{\Omega}$, we can see that **the Euler method, centered at each step, is equivalent to considering that $\mathbf{\Omega}$ is constant over each interval** $[\frac{i}{n}, \frac{i+1}{n}]$. Indeed, the equation $\mathcal{R}' = \mathcal{R}\,\tilde{\mathbf{\Omega}}_i$ with initial condition $\mathcal{R}_i$ may be integrated in

$$\mathcal{R}(s) = \mathcal{R}_i\, exp(s\tilde{\mathbf{\Omega}}_i)$$

We have implemented both Euler methods (centered or not) and since we have not seen any negligible difference, we have chosen to keep the method centered at each step, as this requires fewer calculations. Hence, we have replaced the propagating Euler equation 4 by

$$\mathcal{R}_{i+1} = \mathcal{R}_i\, exp(\frac{1}{n}\tilde{\mathbf{\Omega}}_i) \tag{6}$$

2.3 Interpolation of the frame progression

One could argue that the previous calculation for $\mathcal{R}_t$, based on equation 3, needs to evaluate the angular velocity of the moving frames $\mathcal{R}_0$ and $\mathcal{R}_1$. We present a method that directly interpolates the progression $r_i = \mathcal{R}_i^{-1}\mathcal{R}_{i+1}$ of the moving frame. To interpolate the rotation $r_{t,i}$ from $r_{0,i} = \mathcal{R}_{0,i}^{-1}\mathcal{R}_{0,i+1}$ to $r_{1,i} = \mathcal{R}_{1,i}^{-1}\mathcal{R}_{1,i+1}$ we make use of Shoemake's *slerping* [Sho87] and we will show that this corresponds to a linear interpolation of the angular velocity.

As stated in [BCGH92, HP93], the use of quaternions to represent rotations has several advantages. Quaternions have only 4 coordinates in contrast to the 9 parameters of a rotation matrix. Moreover, there is a simple 2 to 1 mapping from the space of unit-length quaternions to the space of the rotations: the unit-length quaternion $(w, x, y, z) = (w, \mathbf{V})$ is mapped to the rotation

$$r = Id + 2\,w\tilde{\mathbf{V}} + 2\,\tilde{\mathbf{V}}^{\mathbf{2}} \tag{7}$$

Conversely, the rotation of angle α about unit-length axis vector $\mathbf{b}$ corresponds to the quaternion

$$(\cos(\alpha/2),\ \sin(\alpha/2)\,\mathbf{b})$$

and to its opposite. This former mapping is a local isometry: the geodesics around a quaternion are mapped to the geodesics around the corresponding rotation.

If we now choose the corresponding quaternions $q_{0,i}$ and $q_{1,i}$ of $r_{0,i}$ and $r_{1,i}$ such that their scalar product (in $\mathbb{R}^4$) is positive, then $q_{t,i} = slerp(q_{0,i}, q_{1,i}, t)$ will represent the rotation $r_{t,i}$.
In Appendix B, we give an outline of the reasons why the moving frame $\mathcal{R}_t$ obtained by a limiting process (when the sampling step tends to 0) from the equation

$$\mathcal{R}_{t,i+1} = \mathcal{R}_{t,i}\, r_{t,i} \tag{8}$$

corresponds to equation 3, where $\mathbf{\Omega}_t$ is a linear interpolation of $\mathbf{\Omega}_0$ and $\mathbf{\Omega}_1$.

Hence quaternion *slerping* provides a simple way to interpolate the moving frames. On the other hand, no control is allowed over the interpolation of the angular velocity since it is always linear.

3 How to frame a curve

In the previous section, we have described a general interpolation process that roughly amounts to equations 1, 2 and 8. Hence, in order to initialize the process, we wish to evaluate two *adapted* frames $\mathcal{R}_0$ and $\mathcal{R}_1$, given two curves Γ_0, Γ_1.

Below, we discuss two types of adapted frame, point out their relationship with the geometry of the curves and give a construction of these frames.

3.1 The Frenet frame

The Frenet frame [Car76], or FF, is the most commonly used frame. It is characterized by the nullity of the second component of the angular velocity expressed in the local frame. Indeed, if $(\mathbf{t}, \mathbf{f}, \mathbf{g})$ are the axis of the frame $\mathcal{R}$, the angular velocity $\mathbf{\Omega}$ must be of the form $-\lambda\,\mathbf{t} + \mu\,\mathbf{g}$. Identifying $\mathbf{t}'$ and $\mathbf{\Omega} \times \mathbf{t}$ we see that

$$v\kappa\,\mathbf{n} = \mu\,\mathbf{f}$$

thus

$$\mathbf{f} = \pm\mathbf{n} \quad \mathbf{g} = \pm\mathbf{b} \quad \mu = \pm v\kappa \quad \lambda = \pm v\tau$$

where $\mathbf{n}$, $\mathbf{b}$, v, κ and τ are respectively the normal, the binormal, the velocity, the curvature and the torsion of the curve Γ.

Hence, up to the velocity, the components of $\mathbf{\Omega}$ are equal to the curvature and the torsion of the curve.

Now if we interpolate $\mathbf{\Omega_t}$ from $\mathbf{\Omega_0}$ to $\mathbf{\Omega_1}$, keeping the **second** component equal to zero, we will obtain a frame $\mathcal{R}_t$ which is **a FF at each time** and the curvature and torsion of the interpolated curve will be defined by the component of $\mathbf{\Omega_t}$, up to its velocity $v_t = (1-t)v_0 + tv_1$.

For a polyline, a good approximation of the FF may be obtained with an approximated tangent $\mathbf{t}_i$, the approximated binormal $\mathbf{b}_i = (\mathbf{t}_i \times \mathbf{t}_{i+1})/\|\mathbf{t}_i \times \mathbf{t}_{i+1}\|$ and the normal $\mathbf{b}_i \times \mathbf{t}_i$.

3.2 The relatively parallel adapted frame

The relatively parallel adapted frame, or RPAF, is an alternative to the FF. Starting with an arbitrary frame, we propagate the frame along the curve by rotating it as little as possible, while keeping the frame *adapted* to the curve. As stated in [Bis75, Klo86] it has some advantages over the FF. Apart from its minimal rotation property, it is defined at every point for every regular curve, twice continuously differentiable. In particular, the RPAF is defined where the curvature vanishes, although this is not the case for the FF.

From [Bis75], it may be seen that a RPAF is characterized by the nullity of the first component of the angular velocity expressed in the local frame. More precisely their angular velocity takes the form $\mathbf{\Omega} = v\kappa\mathbf{b}$, where v, κ and $\mathbf{b}$ are defined[4] as for the FF.

Now if we interpolate $\mathbf{\Omega}_t$ from $\mathbf{\Omega}_0$ to $\mathbf{\Omega}_1$, keeping the **first** component equal to zero, we will obtain a frame $\mathcal{R}_t$ which is **a RPAF at each time**. Moreover the curvature is given by the norm of the angular velocity $\mathbf{\Omega}_t$, up to the velocity $v_t = (1-t)v_0 + tv_1$.

[4]Note that the product $v\kappa\mathbf{b}$ is defined even if the (bi)normal is not, since, in this case, the curvature κ vanishes.

For a polyline we can construct a good approximation of the RAPF. For this, first note that

$$\lim_{u \to 0} \frac{\arccos < \mathbf{t}(s), \mathbf{t}(s+u) >}{u} = v(s)\kappa(s)$$

Hence, if α_i denotes the angle formed by $\mathbf{t}(\frac{i}{n})$ and $\mathbf{t}(\frac{i+1}{n})$ (see Figure 4), it is seen that $n\alpha_i$ converges uniformly to $v(s)\kappa(s)$. From this and equation 6 we obtain the recursive equation

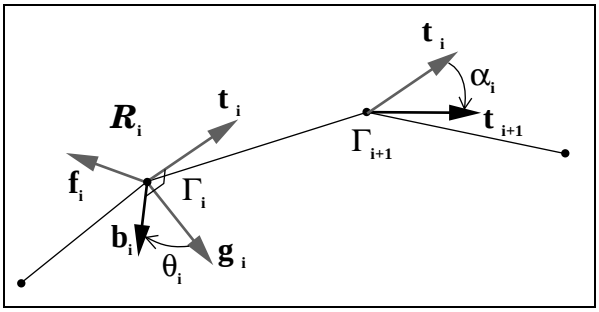

Figure 4: The angles α_i and θ_i.

$$\mathcal{R}_{i+1} = \mathcal{R}_i \, exp(\alpha_i \mathbf{b}_i) \tag{9}$$

So that $\mathcal{R}_{i+1}$ is obtained by rotating $\mathcal{R}_i$ about $\mathbf{b}_i$ of angle α_i. We have thus retrieved the results stated in [Klo86].

4 Interpolation of the angular velocity

In this section, we will discuss the method to interpolate the angular velocity once we have provided an adapted frame for the two curves Γ_0 and Γ_1. This interpolation will determine the geometry of the interpolated curve.

Firstly, we will show how to relate the angular velocity interpolation to the minimization of a deformation measure. Since the geometry of a curve is determined by its curvature κ and its torsion τ, we minimize the length of the path[5] (κ_t, τ_t) that goes from (κ_0, τ_0) to (κ_1, τ_1). This length is given by

$$\mathcal{M}(\Gamma_t) = \int_0^1 \int_0^1 \sqrt{\dot{\kappa_t}(s)^2 + \dot{\tau_t}(s)^2} \, ds dt$$

(the superscript dot accounts for the time derivation). $\mathcal{M}$ is simply minimized by a parametrization of the segment that connects (κ_0, τ_0) to (κ_1, τ_1). For a uniform parametrization

[5] The points of this path should be considered as couples of functions.

we may choose

$$\kappa_t(s) = (1-t)\kappa_0(s) + t\kappa_1(s)$$
$$\tau_t(s) = (1-t)\tau_0(s) + t\tau_1(s)$$

4.1 Link with the frenet frame

For a normal parametrization of a curve provided with the Frenet frame, we have

$$\mathcal{M}(\Gamma_t) = \int_0^1 \int_0^1 \|\dot{\mathbf{\Omega}}_t(s)\|\, ds dt$$

Hence the minimization of $\mathcal{M}$ leads to a linear interpolation of the angular velocity. For our practical needs we choose to keep this interpolation even though the parametrization is not normal. In such case we have

$$\mathbf{\Omega}_t(s) = (1-t)\,\mathbf{\Omega}_0(s) + t\,\mathbf{\Omega}_1(s)$$

so that

$$\kappa_t = \frac{(1-t)\,v_0\kappa_0 + t\,v_1\kappa_1}{v_t} = \kappa_0 \;+\; \frac{1}{(1+\frac{v_0}{v_1}) - \frac{1}{t}\frac{v_0}{v_1}}\,(\kappa_1 - \kappa_0)$$

and

$$\tau_t = \frac{(1-t)\,v_0\tau_0 + t\,v_1\tau_1}{v_t} = \tau_0 \;+\; \frac{1}{(1+\frac{v_0}{v_1}) - \frac{1}{t}\frac{v_0}{v_1}}\,(\tau_1 - \tau_0)$$

Hence,**the linear interpolation of the angular velocity leads to a monotone interpolation of the curvature and of the torsion**.

4.2 Link with the relatively parallel adapted frame

In the case of a linear interpolation of the angular velocity we have

$$\mathbf{\Omega}_t = (1-t)\,\mathbf{\Omega}_0 + t\,\mathbf{\Omega}_1 = (1-t)v_0\kappa_0\mathbf{b}_0 + tv_1\kappa_1\mathbf{b}_1$$

and by the triangular inequality

$$|\kappa_t| \leq |\kappa_0| \;+\; \frac{1}{(1+\frac{v_0}{v_1}) - \frac{1}{t}\frac{v_0}{v_1}}\,(|\kappa_1| - |\kappa_0|)$$

which shows that $|\kappa_t|$ is comprised between $|\kappa_0|$ and $|\kappa_1|$. Actually, we can obtain the same interpolation for the curvature *and* for the torsion as for the Frenet frame. Indeed, in the local frame of a RPAF, the binormal is expressed by (see [Bis75]) $\mathbf{b} = (0, -\sin\theta, \cos\theta)^T$, where θ is a primitive of the torsion (see Figure 4). Thus we have

$$\mathbf{\Omega}_i = n\alpha_i(0, -\sin\theta_i, \cos\theta_i) \tag{10}$$

We obtain a linear interpolation of $v\kappa$ and $v\tau$ by replacing the linear interpolation of $\mathbf{\Omega}_i$ by the linear interpolation of the angles α_i and θ_i.

5 Extensions

We have seen how to interpolate two curves in a rather constrained way. In order to give more flexibility we enable the user to add two types of scalar fields along the curve. One will represent an extra twist of the associated frame, while the other concerns the local speed of the transformation.

5.1 Twisting the frames

In our implementation, the user has the ability to associate a real value to some points of the curve. Those values are then interpolated along the curve and interpreted as angles. Each frame is then rotated about the tangent using these angles. Hence, the associated frame $\mathcal{R}$ (that is the FF or the RPAF) is replaced by

$$\overline{\mathcal{R}} = \mathcal{R}\, r_\varphi$$

where r_φ is a rotation of angle φ about the x-axis. The angular velocity $\overline{\mathbf{\Omega}}$ of $\overline{\mathcal{R}}$ is related to the angular velocity of $\mathcal{R}$

$$\overline{\mathbf{\Omega}} = r_{-\varphi}\, \mathbf{\Omega} - \varphi' \vec{\imath}$$

In order to interpolate two twisted frames $\overline{\mathcal{R}}_0$ and $\overline{\mathcal{R}}_1$ we may either interpolate separately $\mathbf{\Omega}$ and φ, or interpolate $\overline{\mathbf{\Omega}}$ globally. In the first case the geometry of the interpolated curve is independent of the twist, while this is not true in the second case, as may be seen when comparing Figures 11 and 12.

5.2 Control of the transformation speed

The second scalar field is used to control the speed of the transformation. More precisely, we replace in the linear interpolation formula, the time t by a function of the time and of the scalar field so that any quantity x_i will be interpolated with

$$x_{t,i} = (1 - f(t,i))\, x_{0,i} + f(t,i)\, x_{1,i}$$

provided that $f(0,i) = 0$ and $f(1,i) = 1$.

This will change the local rate of the transformation as may be seen from Figure 5.

5.3 Curve straightening

Using a constant twist of angle π, it is possible to straighten any curve with constant length. As a matter of fact, we can transform a curve into one with a twisted angular velocity, using quaternion slerping. As this corresponds to a linear interpolation of the angular velocity, at a half way point, we will have $\mathbf{\Omega}_{\frac{1}{2},i} = 0$. From this and from the fact that the length is linearly interpolated, we see that at time $t = \frac{1}{2}$, the interpolated curve is a straight segment with the same length as the initial curve. Note that the angular velocity follows a geodesic (i.e a

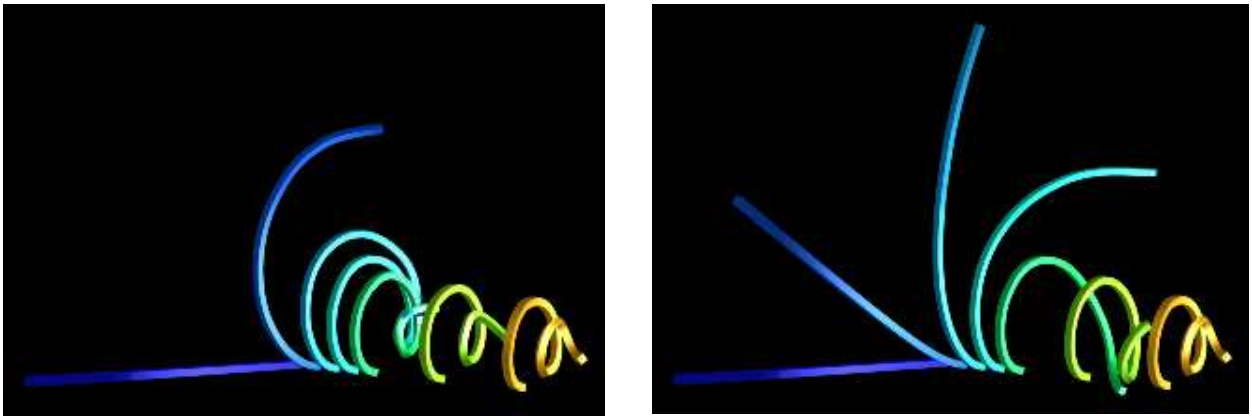

Figure 5: (left) The transformation of a line segment into a spiral. The interpolated curves are visualized at time t = 0, 0.1, 0.2, 0.35, 0.5, 0.75, 1. (right) The transformation of the same curves, but this transformation is gradually accelerated along the initial curve

segment) that passes through 0, hence the transformed path of rotation is also a geodesic in the rotation space. More precisely, as the curve is straightened, the associated moving frame is retracted over the identity following the geodesics (see Figure 6).

6 Implementation and results

In our implementation we have used the RPAF instead of the FF so that we have two modes of interpolation, apart from the extra parameters. One mode uses the quaternion slerping to correspond to a linear interpolation of the angular velocity, as stated in section 2.3, and the other mode uses angles interpolation and is computed with formulas 10, 6, 2 and 1. Below, we will summarize the algorithm steps for both methods. We call the two methods Q and A (for quaternion and angle) respectively.

Preprocessing of Q

For each of the two polylines:

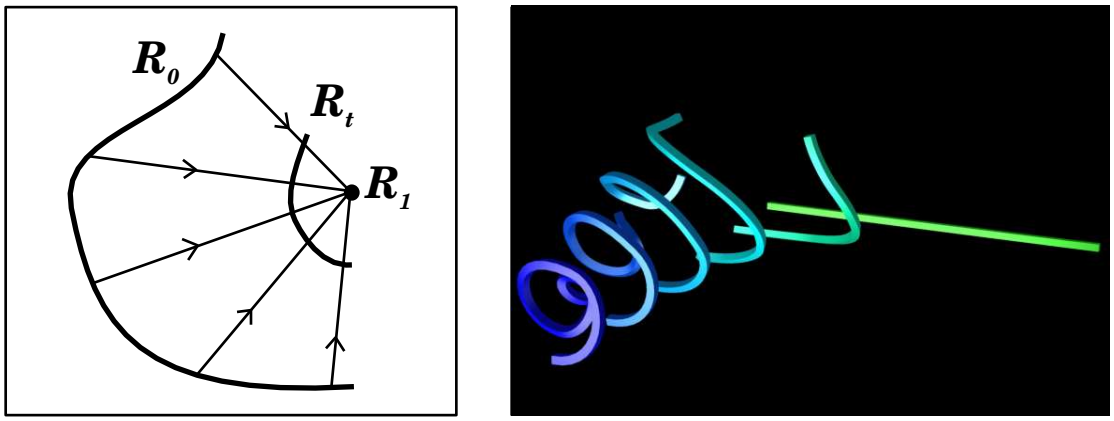

Figure 6: (left) the frame $\mathcal{R}_0$ is retracted over the identity. (right) A straightening of a curve visualized at time t = 0, 0.1, 0.2, 0.3, 0.4, 0.5.

1. **evaluate tangents $\mathbf{t}_i$ at points Γ_i,**
2. **compute the angles $\alpha_i = \angle(\mathbf{t}_i, \mathbf{t}_{i+1})$, the binormales $\mathbf{b}_i = (\mathbf{t}_i \times \mathbf{t}_{i+1})/\|\mathbf{t}_i \times \mathbf{t}_{i+1}\|$, and the quaternions $q_i = (\cos(\alpha_i/2), \sin(\alpha_i/2)\mathbf{b_i})$,**
3. **propagate the frame $\mathcal{R}_0 = (\mathbf{t}_0, \mathbf{f}_0, \mathbf{g}_0) = (\mathbf{t}_0, \mathbf{n}_0, \mathbf{b}_0)$ using equation 9,**
4. **calculate the local coordinates $\gamma_{i+1} = n\mathcal{R}_i^{-1}(\Gamma_{i+1} - \Gamma_i)$.**

Note that, in step 3 of our implementation, we convert the rotations into quaternions for the calculation of the propagation equation 9.

Interpolation at time t with Q

1. **calculate $(\Gamma_{t,0}, \mathcal{R}_{t,0})$ from $(\Gamma_{0,0}, \mathcal{R}_{0,0}), (\Gamma_{1,0}, \mathcal{R}_{1,0})$ and t,**
2. **compute for each i: $q_{t,i} = slerp(q_{0,i}, q_{1,i}, t)$ and $\gamma_{t,i} = lerp(\gamma_{0,i}, \gamma_{1,i}, t)$,**
3. **recover $\Gamma_{t,i}$ using equations 8 (converted in terms of quaternions) and 1.**

Note that step 1 only influences the relative position of the interpolated curve and not its shape. More precisely, the geometry of the transformation is independent of the relative positions of the curves, since we consider the involved entities in local coordinates.

In our implementation we used separately *lerping* (linear interpolation) and *slerping* for the point $\Gamma_{t,0}$ and the frame $\mathcal{R}_{t,0}$.

Preprocessing of A

Same as for Q, and add the calculation of $\theta_i = \angle(\mathbf{g}_i, \mathbf{b}_i)$ in step 4.

Interpolation at time t with A

1. same as for Q,
2. compute for each i: $x_{t,i} = lerp(x_{0,i}, x_{1,i}, t)$, with $x = \alpha, \theta$ or $x = \gamma$,
3. recover $\Gamma_{t,i}$ using equations 10, 6 and 1.

In the following examples we have used sampled cubic splines, and we have swept a square attached to the frames in order to visualize the twist of these frames. In each figure, the dark blue curve is transformed into the yellow one. The same curves are used for Figures 7 to 10. In Figures 7, 8, 9, 11 and 12 we use one spline segment sampled with 40 points, while the curves in Figure 10 have only 10 points. A comparison of Image 8 and 10 shows a very good stability of the algorithm.

In Figures 11 and 12, the initial and final key frame curves have the same shape but the initial curve is twisted (the twist varies from 0 to 4π radian).

7 Conclusion

We have presented a new technique to smoothly transform two polylines, using adapted moving frames. We have also shown the relationship between the two algorithms designed for the polylines and the underlying continuous parametrizations.

Our method is particularly adapted to the control of local geometry (curvature and torsion) as well as of global geometry (length). Through the use of simple intuitive parameters the user can explore a wide variety of transformations. Moreover, the simplicity of the computation allows one to carry out the transformation of curves with several hundreds of points in interactive time. Note that, as a particular feature, our approach leads to a simple way of straightening a curve.

Our technique could be used to animate generalized cylinders, implicit surfaces or any objects defined by a skeleton curve. Future research could focus on the problem of self-intersections, a topic which as not been discussed by the authors in this presentation. In particular, if a curve resembles an open knot, we could investigate how to combine some auto-repulsion of the curve with the evolution process proposed here. To conclude we will borrow the words from Sederberg and al. [SG92]: "of course, extending this algorithm to polygonal surfaces in 3-D is a worthwhile goal".

References

[BCGH92] A.H. Barr, B. Currin, S. Gabriel, and J.F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. *Computer Graphics (SIGGRAPH'92)*, 26(2):313–320, July 1992.

[Bis75] R.L. Bishop. There is more than one way to frame a curve. *American Mathematical Monthly*, (82):246–251, 1975.

[BW76] N. Burtnyk and M. Wein. Interactive skeleton technique for enhancing motion dynamics in key frame animation. *Communications of the ACM*, 19(10):564–569, October 1976.

[Car76] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, New Jersey, 1976.

[Fau89] O. Faugeras. A few steps toward artificial 3D vision. In Michael Brady, editor, *Robotics science*, pages 39–137. MIT press, 1989.

[Fau93] O. Faugeras. Cartan's moving frame method and it's application to the geometry and evolution of curves in the euclidean affine and projective planes. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, number 825, pages 11–46. Springer-Verlag, 1993.

[HP93] G. Hanotaux and B. Peroche. Interactive control of interpolations for animation and modeling. Technical Report STE-93-02, ENSM de Saint-Etienne, Saint-Etienne, France, 1993.

[Klo86] F. Klok. Two moving coordinate frames for sweeping along a 3D trajectory. *Computer Aided Geometric Design*, (3):217–229, 1986.

[KTZ92] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. On the evolution of curves via a function of curvature. I. The classical case. *Journal of Mathematical Analysis and Applications*, 163:438–458, 1992.

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. pages 259–268. ICCV, 1987.

[LCJ94] F. Lazarus, S. Coquillart, and P. Jancène. Axial deformations: an intuitive deformation technique. *Computer-Aided Design*, 26(8):607–613, August 1994.

[LV94] F. Lazarus and A. Verroust. Feature-based shape transformation for polyhedral objects. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[PFTV92] W.H. Press, B.P. Flannery, S.A. Teukolskym, and W.T. Vettering. *Numerical Recipes in C...* Cambridge Univ. Press, Cambridge, England, 1992.

[SG92] T.W. Sederberg and E. Greenwood. A physically based approach to 2-D shape blending. *Computer Graphics (SIGGRAPH'92)*, 26(2):25–34, July 1992.

[SGWM93] T.W. Sederberg, P. Gao, G. Wang, and H. Mu. 2D shape blending: An intrinsic solution to the vertex path problem. *Computer Graphics (SIGGRAPH'93)*, 27(2):15–18, August 1993.

[Sho87] K. Shoemake. Quaternion calculus and fast animation, computer animation: 3-D motion specification and control. In *SIGGRAPH Course Notes*, volume 10, pages 101–121, 1987.

[ST94] G. Sapiro and A. Tannenbaum. Area and length preserving geometric invariant scale-spaces. In Jan-Olof Eklundh, editor, *Computer Vision*, pages 449–458. ECCV'94, Springer-Verlag, 1994.

[Wyv90] B. Wyvill. Metamorphosis of implicit surfaces. In *SIGGRAPH Course Notes*, volume 23, 1990.

A $\Gamma^n_{t,i}$ converges uniformly to Γ_t

With the notation of section 2.1 we must check that

$$\lim_{n\to\infty} \sup_{s\in[0,1]} \|\Gamma^n_{t,\lfloor sn\rfloor} - \Gamma_t(s)\| = 0$$

From the definition of $\Gamma_{t,i}$ and Γ_t we have

$$\|\Gamma^n_{t,\lfloor sn\rfloor} - \Gamma_t(s)\| = \|\Gamma^n_{t,0} + \frac{1}{n}\sum_{k=0}^{\lfloor sn\rfloor-1} \Delta^n_{t,k} - \Gamma_t(0) - \int_0^s \Delta_t(u)\,du\|$$

Transforming the sum into an integral

$$\sum_{k=0}^{\lfloor sn\rfloor-1} \Delta^n_{t,k} = \sum_{k=0}^{\lfloor sn\rfloor-1} \int_{\frac{k}{n}}^{\frac{k+1}{n}} \Delta^n_{t,\lfloor un\rfloor}\,du = \int_0^{\frac{\lfloor sn\rfloor}{n}} \Delta^n_{t,\lfloor un\rfloor}\,du$$

with the fact that $\Gamma^n_{t,0} = \Gamma_t(0)$, we deduce

$$\begin{aligned}
\|\Gamma^n_{t,\lfloor sn\rfloor} - \Gamma_t(s)\| &= \|\int_0^{\frac{\lfloor sn\rfloor}{n}} \Delta^n_{t,\lfloor un\rfloor}\,du - \int_0^s \Delta_t(u)\| \\
&\le \|\int_0^{\frac{\lfloor sn\rfloor}{n}} \|\Delta^n_{t,\lfloor un\rfloor} - \Delta_t(u)\|\,du + \int_{\frac{\lfloor sn\rfloor}{n}}^s \|\Delta_t(u)\|\,du \\
&\le \sup_{s\in[0,1]} \|\Delta^n_{t,\lfloor un\rfloor} - \Delta_t(u)\| + \frac{1}{n}\sup_{s\in[0,1]} \|\Delta_t(s)\|
\end{aligned}$$

We conclude using the fact that $\Delta^n_{t,i}$ converges uniformly to Δ_t, and that Δ_t is bounded over the compact set [0,1].

B The interpolated angular velocity with quaternion slerping

With the notation of section 2.3, let us note

$$q_{0,i} = (\cos(\alpha_{0,i}/2), \sin(\alpha_{0,i}/2)\mathbf{b_{0,i}}) \quad \text{and} \quad q_{1,i} = (\cos(\alpha_{1,i}/2), \sin(\alpha_{1,i}/2)\mathbf{b_{1,i}})$$

Quaternion slerping [Sho87] gives

$$q_{t,i} = Slerp(q_{0,i},\ q_{1,i},\ t) = \frac{q_{0,i}\ \sin(1-t)u_i + q_{1,i}\ \sin tu_i}{\sin u_i}$$

where u_i is the 4 dimensional dot product of $q_{0,i}$ and $q_{1,i}$.
Putting $q_{t,i} = (w_{t,i}, \mathbf{V}_{t,i})$, we have

$$w_{t,i} = \cos(\alpha_{0,i}/2)\frac{\sin(1-t)u_i}{\sin u_i} + \cos(\alpha_{1,i}/2)\frac{\sin tu_i}{\sin u_i}$$

and

$$\mathbf{V}_{t,i} = \sin(\alpha_{0,i}/2)\frac{\sin(1-t)u_i}{\sin u_i}\mathbf{b}_{0,i} + \sin(\alpha_{1,i}/2)\frac{\sin tu_i}{\sin u_i}\mathbf{b}_{1,i}$$

On the other hand, from equation 7, we can show that

$$\lim_{n\to\infty} \frac{r_{t,\lfloor sn\rfloor} - Id}{1/n} = \lim_{n\to\infty} 2\,n\,w_{t,\lfloor sn\rfloor}\,V_{t,\lfloor sn\rfloor} \tag{11}$$

where $r_{t,\lfloor sn\rfloor}$ is the rotation associated to $q_{t,\lfloor sn\rfloor}$. From the expression of $w_{t,i}$ and $\mathbf{V}_{t,i}$, the right member of the former equality is easily shown to converge to $(1-t)\,\mathbf{\Omega}_0(s) + t\,\mathbf{\Omega}_1(s)$, where $n\alpha_{0,i}\mathbf{b}_{0,i}$ (resp. $n\alpha_{1,i}\mathbf{b}_{1,i}$) converge to $\mathbf{\Omega}_0$ (resp. $\mathbf{\Omega}_1$). From this, using classic techniques of numerical analysis, we may conclude that the angular velocity of the interpolated frame, obtained as a limiting process of equation 8, is a linear interpolation of $\mathbf{\Omega}_0$ and $\mathbf{\Omega}_1$.

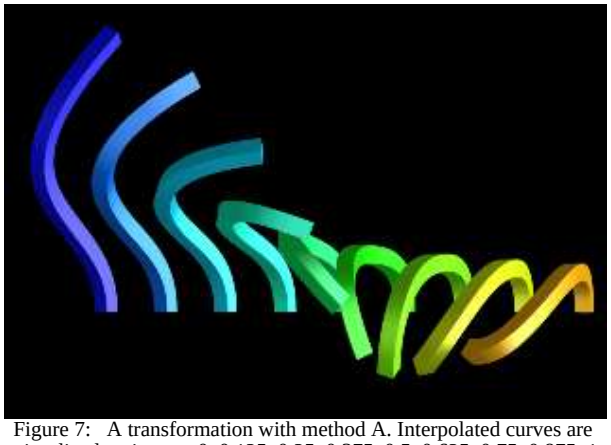

Figure 7: A transformation with method A. Interpolated curves are visualized at time t= 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.

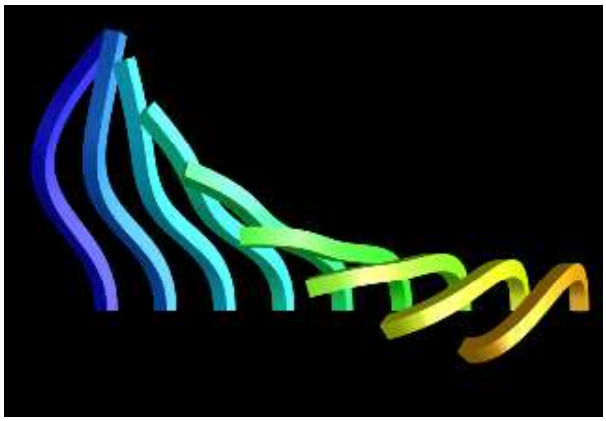

Figure 8: A transformation with method Q. Interpolated curves are visualized at time t= 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.

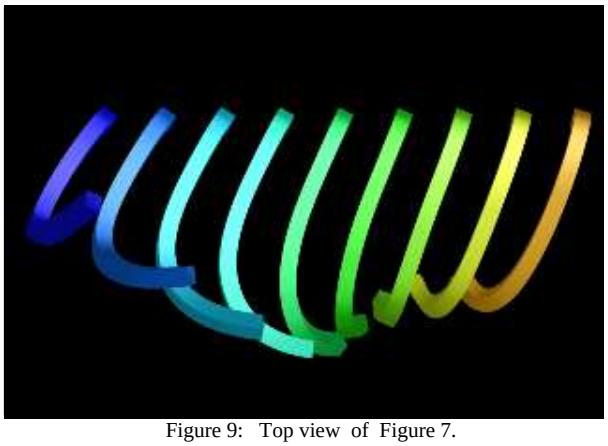

Figure 9: Top view of Figure 7.

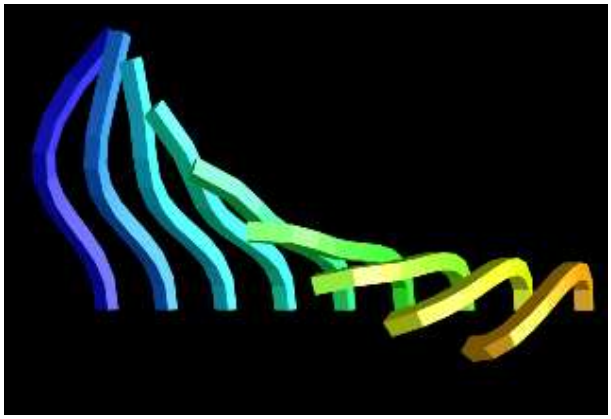

Figure 10: Same as figure 8, but the curves has only 10 points.

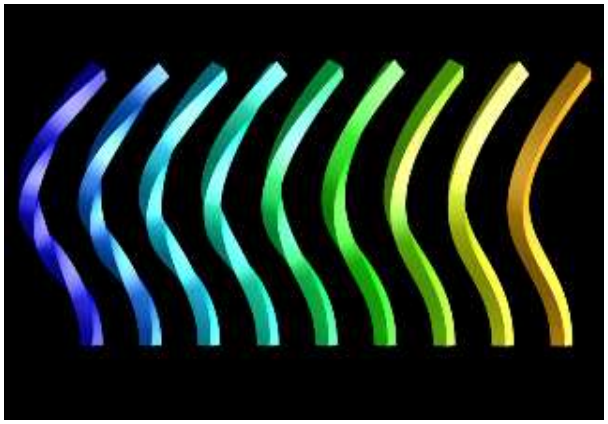

Figure 11: A transformation with method A. The first curve is twisted.

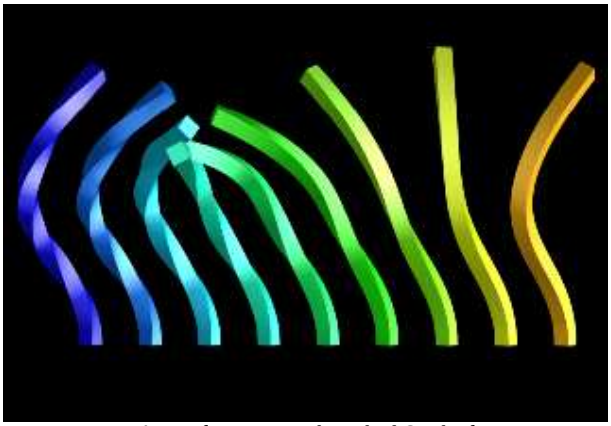

Figure 12: A transformation with method Q. The first curve is twisted as in Figure 11.

Unité de recherche Inria Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 Villers Lès Nancy
Unité de recherche Inria Rennes, Irisa, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Unité de recherche Inria Rhône-Alpes, 46 avenue Félix Viallet, 38031 Grenoble Cedex 1
Unité de recherche Inria Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex
Unité de recherche Inria Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex

Éditeur
Inria, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
ISSN 0249-6399