Haptic Manipulation of Rational Parametric Planar Cubics using Shape Constraints

Christoph Fünfzig*
LE2I, UMR CNRS 5158,
Univ. Bourgogne,
21078 Dijon, France
Christoph.Fuenfzig@u-bourgogne.fr

Philippe Thomin
LAMIH, UMR CNRS 8530,
Univ. Lille Nord de France, UVHC,
59313 Valenciennes, France
Philippe.Thomin@univ-valenciennes.fr

Gudrun Albrecht
LAMAV-CGAO, FR CNRS 2956,
Univ. Lille Nord de France, UVHC,
59313 Valenciennes, France
Gudrun.Albrecht@univ-valenciennes.fr

ABSTRACT

In this paper, we show how to deform a planar rational cubic based on a local interpolation constraint while retaining the qualitative shape of the curve. An impedance-type, parallel haptic device is used to signal changes of the number of inflection points, cusps and loops during the deformation. In this way, the user is provided with an intuitive and natural guidance throughout the curve's shape generation process in CAD.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; J.6 [**Computer-Aided Engineering**]: CAD

Keywords

Haptic Interface, Human-Computer-Interaction, Rational Parametric Cubic, Shape Constraints.

1. INTRODUCTION

Haptic interfaces are becoming more and more popular as human-computer-interaction devices. Such devices have 3-degrees-of-freedom or even 6-degrees-of-freedom for input, and they can give a force feedback to the user.

In the context of *haptic rendering*, the haptic interface enables to explore a model by means of feedback forces [5]. The major difference between haptic rendering and visual rendering lies in the frequency required to give a satisfactory perception to the user. While for the visual rendering, frequencies around 25 Hz are absolutely sufficient, haptic rendering needs about 1 kHz. This implies the necessity of much faster calculations in the case of haptic rendering, which is not always possible due to model complexity. One solution to this problem is to decouple the calculation of the forces from the interactive haptic rendering [5]. In our application, forces are calculated in real time, and we use the NOVINT Falcon parallel haptic device (Figure 1). It has a (4 inch)^3 (approx. (10.16 cm)^3) workspace with 2 lb-capable (approx. 8.9N) actuators and 400 dpi (approx. 157.48 dpcm resolution) sensors.

Haptic devices are very useful tools in the context of rendering geometric constraints. They can be used as new means for assisted input and editing. During input or while editing, the user gets positive hints (by attracting forces) or negative hints (by repelling forces) depending on the state of the current task. The present paper is concerned with haptically assisting the user during the design and the editing process of CAD profile curves according to geometric shape constraints. Related work in the area of haptics for curve manipulations includes Gunn et al. [3] who employ haptic user hints during planning an underground mine made out of lines as basic geometric primitives. Force feedback is used to keep the user away from obstacles in the planning domain and to achieve requirements on the slopes and edges of drawn mine segments. In [7], a haptics-based sketching system is described using cubic spline curves in Bézier form throughout. The haptics simulates canvas friction forces and forces due to a physical curve's mass.

Figure 1: Haptic Devices: the SensAble PHANTOM Omni® is a serial 6DOF input, 3DOF output device (http://www.sensable.com), the NOVINT Falcon® is a parallel 3DOF input/output device (http://www.novint.com). Photos are courtesy of SensAble and NOVINT.

As in [7], planar CAD profiles have traditionally been represented in polynomial cubic B–spline form. But with the more recent NURBS standard *rational* cubic B–Spline curves which are actually composed of several cubic rational Bézier curve segments are more and more used. But unfortunately, in most cases, the additional weight parameters which would be a powerful means of increasing the curve's flexibility are not freely editable by the user. In order to overcome this problem, we propose a haptics-based editing tool for rational CAD profile curves, which on the one hand allow the user to intuitively manipulate the weight parameters, and on the other hand permits user specified shape constraints. We concentrate

*The first author would like to thank the Conseil Régional de Bourgogne for support in a postdoc fellowship.

SAC'10 March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03 ...$10.00.

on one segment of the CAD profile curve, which is represented as a rational cubic Bézier curve in standard form

$$\begin{aligned} & z(t) = \frac{\sum_{j=0}^{3} p_j \omega_j B_j(t)}{\sum_{j=0}^{3} \omega_j B_j(t)} \\ & \text{with } B_j(t) = \binom{3}{j} t^j (1-t)^{3-j}, \text{ the control points } p_j \in \mathbb{R}^2, \\ & \text{and the weights } \omega_j > 0,\ \omega_0 = \omega_3 = 1. \end{aligned} \tag{1}$$

The article [1] provides a shape classification of these curves in terms of its characteristic points. I.e., it describes the locus of inflection points, double points generating loops, and cusps, depending on the position of the control points and on the values of the weights ω_1, ω_2. This two-fold dependency gives rise to the use of two planes, the plane containing the curve (equation (1)), which we call *curve space*, and the plane containing the weights (ω_1, ω_2), which we call *weight space*. The haptic deformation works on both, called *state space*, while moving one control point and two weights.

The paper is organized as follows. Section 2 introduces guiding the user along an existing curve in curve space. This is necessary for an intuitive specification of curve interpolation constraints. Then Section 3 introduces the qualitative shape classification in terms of inflection points, singular points and loops. In Section 4, the direct change of the curve is combined with retaining the qualitative curve's shape. We discuss the different possibilities of setting control points and weights according to deformations at the interactor's position. Finally, Section 5 provides results and conclusions.

2. HAPTIC CURVE EXPLORATION

We use the interactor to guide the user along the curve in curve space so that a curve deformation can be performed at a specific position.

For guiding the user along the curve, we compute a nearest curve point $z(t_x)$ (corresponding to parameter $t_x \in [0,1]$) for the current interactor position x and from that a spring force $f_{spring} = k(z(t_x) - x)$ dragging towards the curve (Figure 2). In order to give strong guidance, the virtual spring should be as stiff as possible, *i.e., $k \in \mathbb{R}^{>0}$* as large as possible. As haptic devices are sampled closed-loop systems [4], they can exhibit instable behavior if the spring constant is chosen too large for the device. To alleviate this problem, we use a piecewise linear spring force. In this way, we can use a large spring constant in the close proximity of the curve and successively smaller spring constants for larger distances. The size of spring constants can be determined experimentally for the used device.

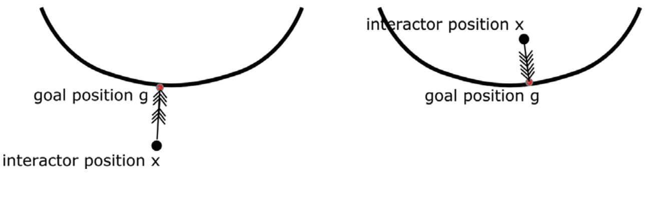


Figure 2: Haptic force employed for curve rendering. A spring force restores the interactor position to the goal position.

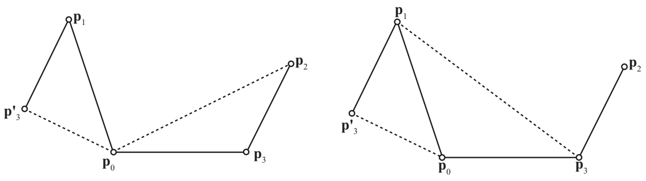
In terms of geometric computations, we need to find a nearest curve point $z(t_x)$ for the current interactor position x. Due to its independence of the curve parameterization, subdivision by the de Casteljau algorithm is a robust algorithm for this problem. It is of branch-and-bound type. Firstly, the curve is subdivided into two curve segments at parameter value $\frac{1}{2}$ via their control polygons. Then, we compute lower bounds for the distance of x to each curve segment geometrically by using the convex hull of the control polygon. In order to consider nearest curve segments first, all curve segments are sorted into a priority queue in ascending order with respect to their lower distance bound. Once the curve segment has become sufficiently short (e.g. is contained in a circle of radius ε), the distance between x and the curve is approximated by the distance between x and an arbitrary point of the control polygon, and the minimum distance found so far is updated. If the lower distance bound is larger than this tentative minimum distance, we can stop the subdivision process, and the tentative minimum distance is the overall minimum. The subdivision algorithm explained above can be implemented so that it is robust and fast at a rate of 1 kHz.

3. SHAPE REGIONS

The shape classification [1] of the rational cubic (equation (1)) identifies shape regions in the (ω_1, ω_2) weight plane corresponding to characteristic points of $z(t)$, such as inflection points, double points, and cusps. The partition of the weight space not only depends on the weights (ω_1, ω_2) but also on the following geometric quantities m and n that characterize the position of the control point p_1 with respect to the coordinate system $\{p_2; p_0 - p_2, p_3 - p_2\}$ in the *curve space*.

$$\begin{aligned} m &= \frac{\det((p_3-p_0),(p_3-p_2))}{\det((p_1-p_0),(p_3-p_2))}, \\ n &= -\frac{\det((p_3-p_0),(p_1-p_0))}{\det((p_1-p_0),(p_3-p_2))}. \end{aligned} \tag{2}$$

These real values m and n have a geometric meaning and represent signed triangle area ratios, where the triangles are defined by the control points of the rational cubic:



$m = \frac{\text{area}(p_0,p_2,p_3)}{\text{area}(p_0,p_1,p_3')}$, $n = \frac{\text{area}(p_0,p_1,p_3)}{\text{area}(p_0,p_1,p_3')}$,
where p_3' is such that $p_3' - p_1 = p_3 - p_2$.

We partition the curve space into six different regions corresponding to the possible positions of the point p_1 with respect to the coordinate system $\{p_2; p_0 - p_2, p_3 - p_2\}$. These regions are characterized by the following values of m and n. Figure 3 provides an illustration of all the regions for control point p_1.

(3)

Region 1. $(m \geq 1$ and $n \geq 1)$ or $(m < 0$ and $n \leq 0)$

Region 2. $m < 0$ and $0 < n < 1$

Region 3. $((m-1)(n-1) < 0$ and $m \neq 0)$ or $(m = 1$ and $n < 1)$ or $(m < 1$ and $m \neq 0$ and $n = 1)$

Region 4. $0 < m < 1$ and $n < 0$

Region 5. $0 < m < 1$ and $0 < n < 1$

Region 6. $0 < m < 1$ and $n = 0$

The curve's behavior is then characterized in weight space by regions that are delimited by the curves $A = 0$, $B = 0$, and $C = 0$, where

$$\begin{aligned}
A &= \omega_1 n^2 - 3\omega_2^2 m(n-1),\ m<0, 0<n<1,\\
B &= \omega_2 m^2 - 3\omega_1^2 n(m-1),\ 0<m<1, n<0,\\
C &= \frac{4m^3(n-1)}{27\omega_1^3} + \frac{4(m-1)n^3}{27\omega_2^3} - \frac{m^2n^2}{27\omega_1^2\omega_2^2}\\
&\quad +(m-1)^2(n-1)^2 - \frac{2mn(m-1)(n-1)}{3\omega_1\omega_2},\\
&\quad m<1, n<1. \qquad (4)
\end{aligned}$$

A region in weight space is referred to as N_i, $i=0,1,2$, if the curve has i inflection points and no singularity (i.e., cusp or double point). It is referred to as L if the curve exhibits a loop. All possible cases are illustrated in Figure 3. The resulting classification is summarized below for the case where the points p_0, p_2, and p_3 are not collinear.

In region 1 $z(t)$ has neither inflection points nor singularities.

In region 2 $z(t)$ has two inflection points, if $C<0$,
a cusp, if $C=0$,
a double point, if $C>0$ and $A>0$,
neither a singular point nor an inflection point, if $C>0$ and $A\leq 0$.

In region 3 $z(t)$ has an inflection point.

In region 4 $z(t)$ has two inflection points, if $C<0$, (5)
a cusp, if $C=0$,
a double point, if $C>0$ and $B>0$,
neither a singular point nor an inflection point, if $C>0$ and $B\leq 0$.

In region 5 $z(t)$ has 2 inflection points, if $C<0$,
a cusp, if $C=0$,
a double point, if $C>0$.

In region 6 $z(t)$ has 2 inflection points, if $C<0$,
a cusp, if $C=0$,
a double point, if $C>0$.

If the three control points p_0, p_2, p_3 are collinear, we exchange the roles of points p_1 and p_2, and apply the cases for regions 1, 3, 6 similarly.

Similar to the curve exploration in curve space (Section 2), we can render the weight space of the curve. The two variable weights (ω_1,ω_2), $\omega_1\geq 0$, $\omega_2\geq 0$ represent a point in the weight space, for which the classification in Figure 3 gives the corresponding shape properties of the curve. For exploring weight space, we move the point in weight space and classify it according to the regions shown in Figure 3. For this task, it is sufficient to compute the values m, n in equation (2) and A, B, and C in equation (4). In regions with special properties, *i.e.,* regions 2, 3, 4, 5 and 6, we use different haptic textures [8] on the weight plane. While moving the point (ω_1,ω_2) on the weight plane, the haptic interactor performs a characteristic movement in orthogonal, physical direction to the weight plane. In this way, the user can be informed about the current shape properties concurrently to other tasks. For example in Figure 4, right, the three different shape properties (loop, inflection points, no special property) are distinguished by two characteristic haptic textures, and no texture in case of no special property.

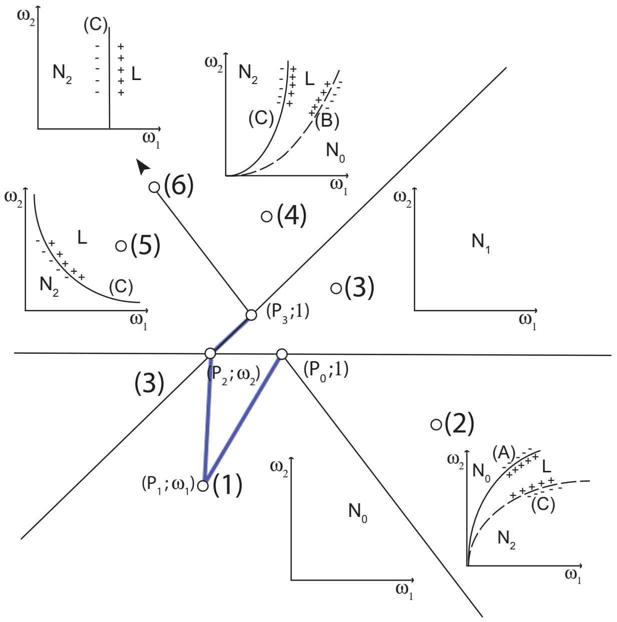
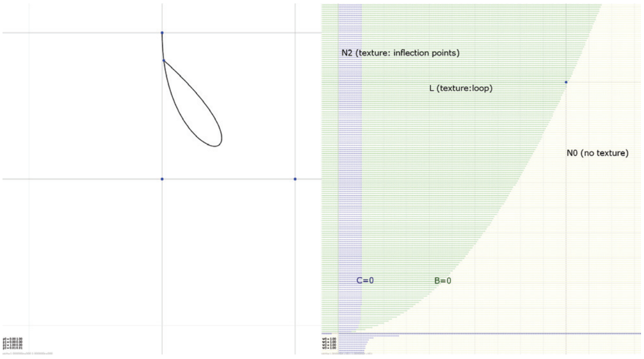


Figure 3: Shape regions in the weight plane (ω_1,ω_2) for all positions of the control point p_1 relative to the fixed control points p_0, p_2, p_3. The symbols $+$ and $-$ indicate the signs of the quantities $A(\omega_1,\omega_2)$, $B(\omega_1,\omega_2)$, $C(\omega_1,\omega_2)$ on each side of the curves $A=0$, $B=0$, $C=0$. N_i denotes a region with $i=0,1,2$ inflection points, and L denotes a region with a loop. A cusp occurs on the curve $C=0$.



curve space weight space

Figure 4: Illustration of the graphical interface: curve space (left) showing the current form of the rational cubic, weight space (right) visualizing the different texture regions.

4. CURVE DEFORMATION IN STATE SPACE

The benefit of haptic interaction is that it allows a direct manipulation in state space. Instead of changing control points and weights separately, we would like to deform the curve at the current interaction point directly. We have to solve the following interpolation problem

$$z(t') = x' \qquad (6)$$

where t' is the curve parameter of the deformation,
and x is the old interaction point,
and x' is the new interaction point.

The parameter t', where the deformation occurs, is chosen at the beginning of the deformation sequence by pressing a button of the interactor. The curve point at this time is $x = (x_0, x_1)$, which is dragged to point $x' = (x'_0, x'_1)$ subsequently.

For a rational parametric cubic, there are several possibilities to solve the interpolation problem. In [2], solving an interpolation constraint for curves defined by arbitrary bases is discussed. The use of rational bases is possible, although the curve's weights do not change during manipulation. In [6], the curve's changes are studied while changing a single weight of a rational B-spline curve for a fixed parameter. Changes of the Bézier curve with this approach are always in the direction to one of the control points p_1 or p_2, a situation which is difficult to achieve for arbitrary deformations.

In Section 4.3, we detail our approach for solving the interpolation constraint through a point x'. As this might not be possible while retaining the same qualitative shape properties, we move the interpolation point $x'(s)$, $s \in [0,1]$ on the line segment xx' in a binary search. For an intermediate interpolation point $x'(s)$, we try to find an interpolating curve by changing one control point (Section 4.1) and the two inner, non-negative weights (Section 4.2). If this is not possible while retaining the same qualitative shape properties, we generate a resistance force proportional to the difference $x'(s) - x'$, which could not be realized.

4.1 Changing only control points

The interpolation problem in equation (6), changing only control points, can be solved as an under-determined linear system for the new control points p'_j

$\sum_{j=0}^{3} \delta p_j \frac{\omega_j B_j(t')}{\sum_{k=0}^{3} \omega_k B_k(t')} = x' - x$

where $\delta p_j = p'_j - p_j$ is the difference between the new control point p'_j and the old control point p_j.

The system matrix $B = \left(b_{ij} := \frac{\omega_j B_j(t')}{\sum_{k=0}^{3} \omega_k B_k(t')} \right)_{i=0,1,j=0,\ldots,3}$ has two rows and four columns for control points p_j in 2-space. As noticed in [2], the solution can be written in terms of orthogonal components $\delta p = B^t \lambda + z$, where $B^t \lambda$, $\lambda \in \mathbb{R}^2$ is in the row space of B, and $z \in \mathbb{R}^4$ is in the null space of B, *i.e.*, $Bz = 0$. For the *minimum-length solution* δp, z must be chosen $z = 0$, which results in $\lambda = (BB^t)^{-1}(x' - x)$ and $\delta p = B^t(BB^t)^{-1}(x' - x)$. In the minimum-length solution δp, all components δp_j can be non-zero.

Our way to handle the under-determined system is to change only one control point $p_{j'}$, having largest influence at the parameter t'. Then it is $\delta p_j = 0$ for $j \neq j'$. The weights keep their current values similar to a non-rational Bézier curve.

We have to solve the following system for $\delta p_{j'}$

$$\delta p_{j'} \omega_{j'} B_{j'}(t') = (x' - x) \sum_{k=0}^{3} \omega_k B_k(t') \qquad (7)$$

4.2 Changing only weights

Similarly, we can solve the interpolation problem in equation (6) by changing the two inner weights ω_1, ω_2. In this case, we keep all control points at their current values, and solve for the weights.

After reordering equation (6) with components $i = 0, 1$ into the form

$$\begin{array}{rcl} \left.\begin{array}{r} \omega'_1(B_1(t')p_{1,0} - B_1(t')x'_0) \\ +\omega'_2(B_2(t')p_{2,0} - B_2(t')x'_0) \end{array}\right\} & = & \left\{\begin{array}{r} B_0(t')x'_0 + B_3(t')x'_0 \\ -B_0(t')p_{0,0} - B_3(t')p_{3,0} \end{array}\right. \\ \left.\begin{array}{r} \omega'_1(B_1(t')p_{1,1} - B_1(t')x'_1) \\ +\omega'_2(B_2(t')p_{2,1} - B_2(t')x'_1) \end{array}\right\} & = & \left\{\begin{array}{r} B_0(t')x'_1 + B_3(t')x'_1 \\ -B_0(t')p_{0,1} - B_3(t')p_{3,1} \end{array}\right. \end{array} \qquad (8)$$

we have to solve for the two inner weights $\omega'_1 = \omega_1(x')$, $\omega'_2 = \omega_2(x')$. For an arbitary point x' to be interpolated, one or both of the weights can become negative. So this problem is only well-posed if the point x' is in the convex hull of the control points p_0, p_1, p_2, p_3.

4.3 Changing control points and weights

In the case of moving one of the interpolated, end control points p_0 or p_3, we always do interpolation by changing only this control point as in Section 4.1. Let us now consider the case of moving the control point p_1, where control points p_0, p_2, p_3 are fixed. The case of moving the control point p_2 is similar.

In a binary search, we move the interpolated point along the line segment $x'(s) = x + s \cdot (x' - x)$, $s \in [0,1]$. Remember that the binary search maintains a parameter interval $[s_{\min}, s_{\max}]$ and continues with interval $[s_{\min}, s_{\text{mid}}]$ or $[s_{\text{mid}}, s_{\max}]$ based on the mid point $s_{\text{mid}} := 1/2(s_{\min} + s_{\max})$.

In a second binary search, we try to find an interpolating curve by choosing p_1 on the line segment $p'_1(u) = p_1 + up_1(x'(s))$, $u \in [0,1]$, where $p'_1(0) = p_1$ and $p'_1(1) = p_1(x'(s))$ is the control point according to Section 4.1. Using the control point $p'_1(u)$, we compute the weights $\omega_1(x'(s))$, $\omega_2(x'(s))$ according to Section 4.2. For $u_{\min} := 0$, the interpolation condition must be fulfilled by changing the weights $\omega_1(x'(s))$, $\omega_2(x'(s))$ alone, and for $u_{\max} := 1$, the interpolation condition is fulfilled with control point $p_1(x'(s))$ and the initial weights ω_1, ω_2. As we want to keep the convex hull property of the curve, we have to retain non-negative weights $\omega_1(x'(s))$, $\omega_2(x'(s))$, which is the case in a neighborhood of $u_{\max}$. If the weights $\omega_1(x'(s))$, $\omega_2(x'(s))$ for u_{mid} are negative (or outside a given interval $[\min_\omega, \max_\omega]$, $\min_\omega \geq 0$), we continue with the search interval $(u_{\text{mid}}, u_{\max}]$. Otherwise, we continue with the search interval $[u_{\min}, u_{\text{mid}}]$.

In the final curve state, interpolating $x'(s)$, we evaluate the expressions A, B, C of the shape classification in Section 3. If the signs of A, B, C change, the curve's characteristics change also, and we continue with the search interval $[s_{\min}, s_{\text{mid}}]$ to keep the curve's characteristics. Otherwise, we continue in the search interval $[s_{\text{mid}}, s_{\max}]$. Notice that for the same state in table (5), $p'_1(x')$ is not allowed to change the region relative to p_0,p_2,p_3. When it changes region, we have to do a reclassification according to tables (3) and (5). This results in a segmentation of the deformation into several steps, which can be performed initially on the line segment $p_1 p_1(x')$.

The search ends in a parameter s' with a curve of same shape properties, interpolating a point $x'(s')$ between x and x'. In the case the user interpolation request $x(t') = x'$ can not be satisfied, we generate a resistance force from the difference $x'(s') - x'$ between the realized and the requested value.

Figure 5 shows a deformation sequence hinted by the haptic interactor. Here, the deformation transforms the curve with inflection point into one without an inflection point. The red line segment in the figure shows the interpolation points input with the haptic interactor by the user, and the resulting movement of control point p_2 along the blue line segment until it passes through the line $p_1 p_3$. There the user is stopped as the curve changes its current properties.

The deformation can also be formulated as a system of constraints on the control point p'_1 and weights (ω'_1, ω'_2). Let $A_s := \text{sgn}(A(\omega_1, \omega_2))$, $B_s := \text{sgn}(B(\omega_1, \omega_2))$ and $C_s := \text{sgn}(C(\omega_1, \omega_2))$ be the current signs of the functions A, B, C. Let s_1, s_2, s_3 in $\{-1, 0, 1\}$ be the signs $s_1 := \text{sgn}((p_1 - p_0)(p_2 - p_0)^{\perp})$, $s_2 := \text{sgn}((p_1 - p_2)(p_3 - p_2)^{\perp})$, $s_3 := \text{sgn}((p_1 - p_3)(p_0 - p_3)^{\perp})$, $v^{\perp} := (-v_1, v_0)$, given by the location of p_1 relative to p_0 p_2, p_2 p_3,

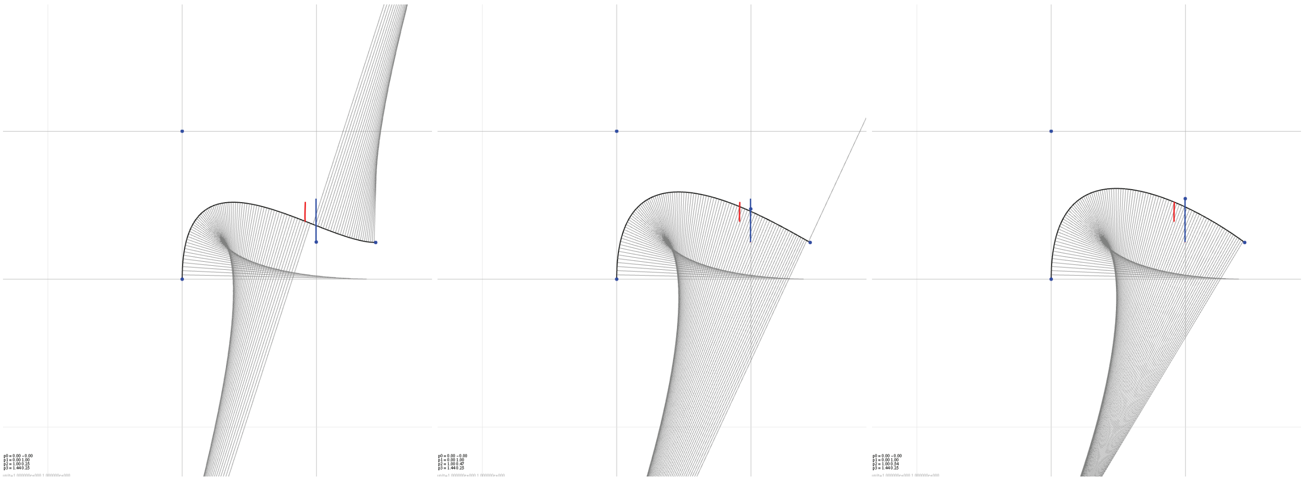


Figure 5: Interpolation sequence hinted by the haptic interactor (left, red line segment), and the blue line segment sketches the corresponding control point change. Interpolation is not possible without removing the inflection point (middle), which is signaled by a resistance force at the interactor. After releasing and setting up again, the full interpolation can be achieved with a curve without an inflection point (right). Visually, the location of the inflection point can be identified by the side change of the grey curvature vectors with respect to the curve.

$p_3\ p_0$. In the following, we use $\mathrm{o}_s \equiv \begin{cases} < & \text{for } s=-1 \\ = & \text{for } s=0 \\ > & \text{for } s=1 \end{cases}$

$$
\begin{array}{lll}
\multicolumn{3}{l}{\text{For components } i=0,1:} \\
\left.\begin{array}{l} \omega'_1(B_1(t')p'_{1,i}-B_1(t')x'_i) \\ +\omega'_2(B_2(t')p_{2,i}-B_2(t')x'_i) \end{array}\right\} & = & \left\{\begin{array}{l} B_0(t')x'_i+B_3(t')x'_i \\ -B_0(t')p_{0,i}-B_3(t')p_{3,i} \end{array}\right. \\
A(\omega'_1,\omega'_2) & \mathrm{o}_{A_s} & 0 \\
B(\omega'_1,\omega'_2) & \mathrm{o}_{B_s} & 0 \\
C(\omega'_1,\omega'_2) & \mathrm{o}_{C_s} & 0 \\
(p'_1-p_0)(p_2-p_0)^{\perp} & \mathrm{o}_{s_1} & 0 \\
(p'_1-p_2)(p_3-p_2)^{\perp} & \mathrm{o}_{s_2} & 0 \\
(p'_1-p_3)(p_0-p_3)^{\perp} & \mathrm{o}_{s_3} & 0 \\
\min_\omega \quad \leq \quad \omega'_1 & \leq & \max_\omega \\
\min_\omega \quad \leq \quad \omega'_2 & \leq & \max_\omega
\end{array}
$$

Note that this system with 4 variables (ω'_1, ω'_2, $p'_{1,0}$, $p'_{1,1}$) and 2 equations is under-constrained. The domain of variables ω'_1, ω'_2 is bounded from below by $\min_\omega \geq 0$ and can also be bounded from above by $\max_\omega \in \mathbb{R}$. The described bisection search restricts the variables p'_1 to the line $p_1\ p_1(x')$ and favors changing the weights (ω'_1, ω'_2) over changing the control point p'_1.

5. CONCLUSION

The presented approach combines a geometric change of the rational curve's shape, specified by an interpolation condition, with qualitative constraints (like same number of inflection points, double points, and cusps) of the curve's shape. Especially, changes of these qualitative properties are signaled by a resistance force at the haptic device so that the user is aware of them. We have integrated the haptics-based deformation into a combined graphical editor for 2D profile curves defined by rational cubic Bézier segments. In Figure 4 the graphical interface is shown with the curve space on the left side and the weight space on the right side. The view of the weight space is optional, and the user can choose to only feel the curve's properties via overlayed haptic textures.

Our constraint solving approach is based on bisection (Section 4.3), which is very fast and makes it possible to compute continuous resistance forces.

In terms of applications, the presented approach can be readily integrated into a CAD system for intuitively constructing and editing 2D rational curves according to shape constraints. Changes of the Bézier curve $z(t)$ occur for all parameters t, and are thus *global changes*. The technique can be used on a Bézier segment of a larger rational B–spline curve for local changes.

As future work, it might be interesting to explore the under-constrainedness of the constraint system further, and to consider changes of the control point p_1 away from the line segment $p_1\ p_1(x')$.

6. REFERENCES

[1] G. Albrecht, J. Bécar, and X. Xiang. Géométrie des points d'inflexion et des singularités d'une cubique rationnelle. *Revue Electronique Francophone d'Informatique Graphique*, 2(1):33–46, 2008.

[2] B. Fowler and R. Bartels. Constraint-based curve manipulation. *IEEE Comput. Graph. Appl.*, 13(5):43–49, 1993.

[3] C. Gunn and P. Marando. Experiments on the haptic rendering of constraints: Guiding the user. In *Proc. Advanced Simulation Technology and Training Conf.*, 1999.

[4] V. Hayward and K. MacLean. Do it yourself haptics, part-1. *IEEE Robotics and Automation Magazine*, (4):88–104, 2007.

[5] S. D. Laycock and A. M. Day. A survey of haptic rendering techniques. *Comput. Graph. Forum*, 26(1):50–65, 2007.

[6] L. Piegl. Modifying the shape of rational b-splines. part 1: curves. *Comput. Aided Des.*, 21(10):509–518, 1989.

[7] C. Raymaekers, G. Vansichem, and F. van Reeth. Improving sketching by utilizing haptic feedback. In *American Assoc. for Artificial Intelligence Spring Symposium (AAAi 2002)*, pages 113–117, March 25-27 2002.

[8] J. Shopf and M. Olano. Procedural haptic texture. In *Proc. UIST Symposium*, pages 179–186, New York, NY, USA, 2006. ACM.