Stable Inverse Dynamic Curves

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Joëlle Thollot

Stable Inverse Dynamic Curves

Alexandre Derouet-Jourdan Florence Bertails-Descoubes Joëlle Thollot

INRIA / LJK, Grenoble University, France

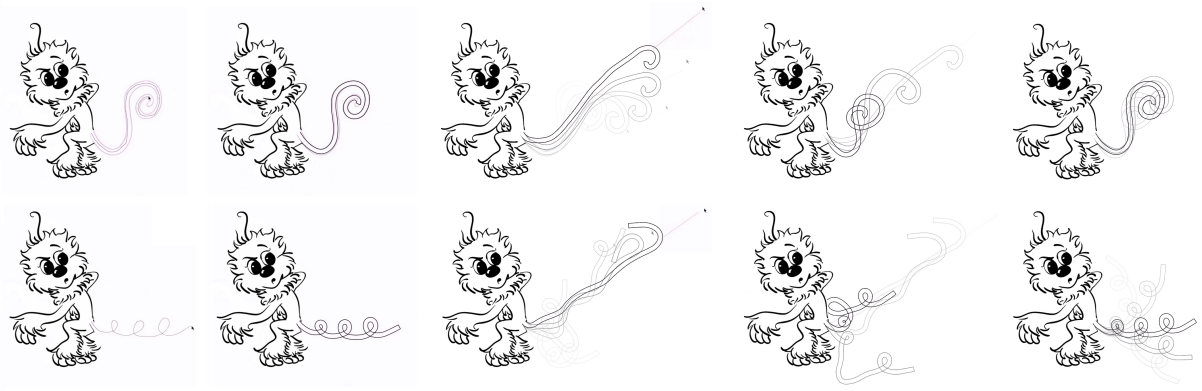


Figure 1: *From left to right: The user sketches a smooth curve over the tail of a character. The curve is automatically converted into a dynamic rod model at stable equilibrium under gravity. The user can then animate the curve (*e.g.*, pull then release it) with the guarantee that the chosen initial shape will be preserved after slight (or possibly strong) motion. See the accompanying video for the full animation.*

Abstract

2d animation is a traditional but fascinating domain that has recently regained popularity both in animated movies and video games. This paper introduces a method for automatically converting a smooth sketched curve into a 2d dynamic curve at stable equilibrium under gravity. The curve can then be physically animated to produce secondary motions in 2d animations or simple video games. Our approach proceeds in two steps. We first present a new technique to fit a smooth piecewise circular arcs curve to a sketched curve. Then we show how to compute the physical parameters of a dynamic rod model (super-circle) so that its stable rest shape under gravity exactly matches the fitted circular arcs curve. We demonstrate the interactivity and controllability of our approach on various examples where a user can intuitively setup efficient and precise 2d animations by specifying the input geometry.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: 2d curve, inverse statics, elastica, stable equilibrium

1 Introduction

2d computer imagery is used in many applications ranging from industrial design to video games and animation feature films. For building intuitive, interactive and creative 2d worlds, much work has been devoted to the geometric design of 2d shapes. Recently, the *dynamics* of 2d objects has gained an increasing interest and now calls for intuitive, fast and semi-automatic solutions for moving and deforming geometry. In this paper, we focus on the *inverse coupling problem* consisting in imposing a geometric rest shape to a physics-based simulator. We specifically handle the widespread case of a rest shape defined as a planar smooth curve.

Our attention has first been drawn towards 2d video games as these are one of the first 2d applications to use dynamics for generating motion from the geometry input created by the game players. Recent video games such as *The World of Goo* (www.worldofgoo.com) or *Algodoo* (www.algodoo.com) specifically set the core of their gameplay around this coupling between geometry and physics. The representation of deformable objects remains however limited (*e.g.*, no flexible primitive for matching a stroke). More importantly, the geometry-physics coupling simply consists in initializing the physics-based models with a given shape *in the absence of any surrounding force*. This assumption does not, in general, reflect the intention of the user when sketching a shape in a given environment. He/she would rather spontaneously model a shape *under gravity*.

In 2d animation movies, motion and deformations are often processed through mere geometric transformations such as interpolation, morphing or warping, *e.g.* [Alexa et al. 2000; Baxter et al. 2009]. In the context of 3d animation, Barzel [1997] developed a parametric rod system dedicated to keyframe animation, providing a direct but limited control over rest shapes and deformations. Though practical, such a descriptive model obviously loses the richness and versatility offered in contrast by a *dynamic* model. Wither *et al.* [2007] set up a sketching interface for initializing a dynamic strand simulator with approximate curvatures computed in the absence of any surrounding force. To the best of our knowledge, Hadap [2006] was the only one to propose a technique – based on inverse statics – that allows for precise geometric control over the rest shape of dynamic strands. Still, the proper matching of arbitrary curly shapes remains difficult and no analysis of equilibrium stability was proposed.

This paper brings a solution to the *stable* geometry-physics coupling problem for 2d smooth lines with straight or curly geometry. Such 1d primitives naturally emerge from simple sketches while allowing for the representation of a wide class of 2d objects ranging from wires and ropes to hair, trees, character skeleton, etc. Our method automatically converts the input curve into the *stable equilibrium* shape of a dynamic rod model, in the presence of external forces such as gravity. As we target video games and interactive 2d animation prototyping, the user should intuitively choose his/her trade-off between speed *vs.* accuracy all along the process. To match all these requirements, we based our work on the dynamic super-helix model [Bertails et al. 2006] restricted here to 2d and referred to as the *super-circle* model in the remainder of the paper.

2 Related work

Representing arbitrary 2d curves as compact and analytic primitives has been an active area of research for more than half a century, originally motivated by curve and surface design for engineering applications. As a result a large amount of curve primitives, ranging from parametric polynomial representations (splines) to piecewise circular arcs and clothoids, are now commonly used to represent 2d curves. The choice for one or another primitive is essentially guided by geometric concerns such as interactive editability, global or local user control, interpolation of original point/tangents, fairness quality, data compression rate *w.r.t.* the input curve, etc.

In contrast, our choice is motivated by both geometric *and* animation concerns: we want to initialize a fast and realistic dynamic curve model with an imposed *arbitrary smooth* curve (possibly with loops) as a stable rest shape. These requirements have an impact on the choice of both the dynamic rod simulator and the geometric primitive for representing the input curves.

Inverse dynamics for rods The dynamics of 2d curves can be naturally achieved by considering the mechanics of thin inextensible rods subject to bending. Various discrete models have been proposed in graphics for 3d inextensible rods with bending and twisting elasticity. While many of them rely on *maximal coordinates* formulations [Spillmann and Teschner 2007; Spillmann and Teschner 2008; Theetten et al. 2008; Bergou et al. 2008; Selle et al. 2008], others prefer to use *reduced coordinates* models because they offer a compact and intuitive parameterization of the rod [Bertails et al. 2006; Hadap 2006; Bertails 2009]. Using a reduced chain of articulated bodies as a rod model, Hadap [2006] was the only one to propose a method for fitting a rod at rest under gravity to a given input curve. The static fitting is performed by *inverse dynamics*, a well-known principle in robotics consisting in computing the joint forces and torques of a kinematic chain given the reduced accelerations and velocities of the joints [Featherstone 1983]. Hadap highlights that inverse dynamics would be difficult to achieve using a maximal coordinates rod model, for which the kinematics of the rod has to be explicitly enforced through penalty terms or external constraints. We fully support this analysis.

We depart from Hadap's approach by using an alternative reduced coordinates model. In general, a serial chain of rigid links is not adapted to capture the geometry of arbitrary complex curves possibly with multiple loops. Such shapes require lots of rigid links to match a complex input geometry. They furthermore imply the retrieval of strong joint torques which may lead to numerical instabilities. Another interesting reduced coordinates model is the super-helix model [Bertails et al. 2006], characterized by high-order elements that are smoothly connected – helices in 3d and circular arcs in 2d. With a small set of intuitive parameters, this model can represent complex geometric shapes and capture in real-time typical deformations of an elastic inextensible rod, with real-time performance for good resolutions (up to 50 elements in 2d). For these reasons we chose this model – referred to as *super-circle* in the remainder of this paper – for solving our inverse problem. The price to pay will simply consist in building an algorithm for approximating a smooth curve with a controlled number of circular arcs. Controlling the number of arcs at this stage is crucial as it will set the resolution of the animation model and hence the computational load of subsequent simulation whatever the chosen animation algorithm (quadratic [Bertails et al. 2006] or linear [Bertails 2009] in time). We strongly believe that offering direct control over the resolution of a simulator is of great interest because it allows for a precise tuning between quality and time performance at the different stages of the animation design.

Compared to [Hadap 2006] we also provide an analysis of *stability* of the fitted rod's configuration and a simple algorithm to guarantee that the initial configuration is stable. This feature is desirable for an artist in order to have a given shape mostly restored after slightly (or even strongly) deforming it.

Piecewise circular arcs approximation Approximating curves by piecewise circular arcs has been extensively studied in computer-aided design, see *e.g.* [Yang and Du 1996; Pei and Horng 1996; Horng 2003; Safonova and Rossignac 2003; Drysdale et al. 2008]. More recently, piecewise clothoids, made of arcs with linear (instead of constant) curvature, have gained an increasing interest in the graphics community [McCrae and Singh 2008; Baran et al. 2010] because of their fairness property, allowing for sketching visually pleasing curves.

Previous approaches however do not fully account for our three main goals: (a) the *quasi-instantaneous* geometric fitting, (b) the *precise control* over the curve resolution and c) the fitting of a physics-based rod model to a *smooth* curve. Typically most authors prefer to control a precision error instead of the number of arcs [Yang and Du 1996; McCrae and Singh 2008; Baran et al. 2010], while others do not enforce G^1 smoothness [Pei and Horng 1996]. Furthermore, most previous approaches are far from interactive. The only ones, including ours, that fulfill this requirement are based on an automatic pruning of the input points to keep only a small set of *gate* points before applying the reconstruction [Pei and Horng 1996; McCrae and Singh 2008].

3 Contributions

An overview of our method is presented in figure 2. Our contributions are twofold:

- We introduce a new algorithm for accurately and interactively fitting a G^1-smooth piecewise circular arcs curve to any 2d smooth curve, with exact user control over the number of arcs (shall it be odd or even). Our method relies on dynamic programming for segmenting the curve into a predefined number of regions optimally matched by circular arcs, combined with an original reconstruction method based on floating tangents interpolation. The advantages of our method are enhanced by providing thorough comparisons to previous circular arcs fitting methods used in graphics and computer-aided design.
- We present a simple and efficient algorithm for finding the parameters of the super-circle model so that its rest shape under gravity exactly matches a given G^1-smooth piecewise circular arcs curve. We show that the super-circle model is perfectly adapted to inverse dynamics and we provide a mathematical proof for the existence of solutions to the stable inverse dynamic problem. To the best of our knowledge, the question of managing the stability of equilibrium is new to graphics, and greatly enriches the set of existing tools for controlling physics-based animations.

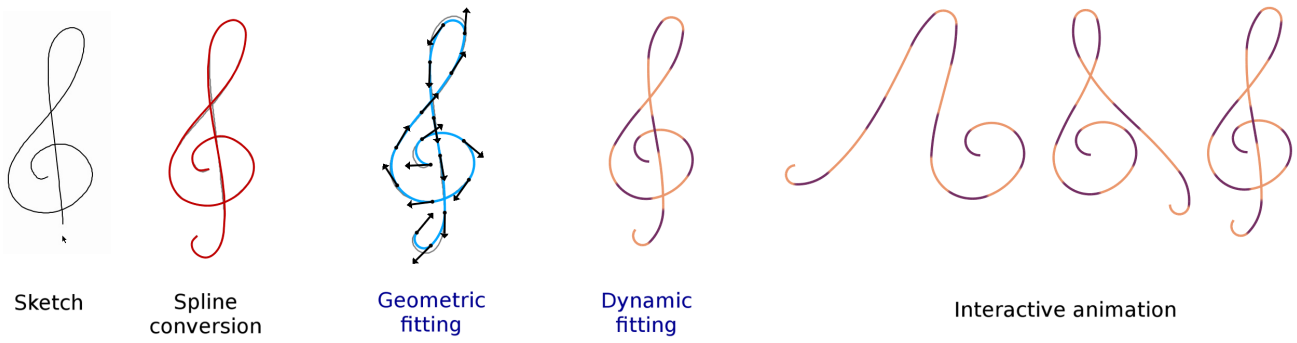
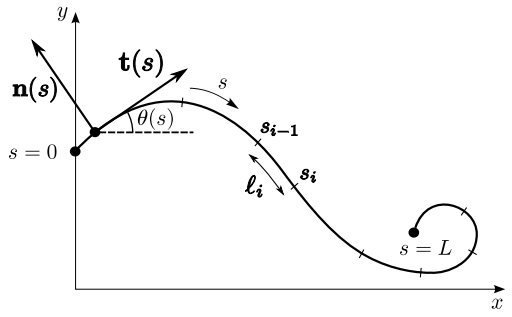


Figure 2: *Overview of our method.*

Our paper is organized as follows. We first recall the main notations and equations for a super-circle in Section 4. Sections 5 and 6 present our two major contributions. Finally, results and applications are presented in Section 7 before concluding.

4 Notations for a super-circle

Let us consider an inextensible elastic rod with one clamped end and the other end free. Let $\mathbf{r}(s)$ denote the centerline of the rod, parameterized by the curvilinear abscissa s. In the super-helix model, material curvatures and twist are assumed to be piecewise constant along the rod, leading to a piecewise helical centerline [Bertails et al. 2006]. In 2d, this discrete model boils down to a material curve with a piecewise constant curvature $\kappa(s)$ (no twist), *i.e.*, a centerline made of N piecewise circular arcs. Let $\theta_0 = \theta(0)$ be the clamping angle of the rod and L its total length. The material frame $\{\mathbf{t}(s), \mathbf{n}(s)\}$ continuously slides along the centerline through infinitesimal rotations around the z axis (clock counter-wise when the curvature is positive).



Let κ_i be the constant curvature of the i^{th} element of the rod, κ_i^0 its natural curvature, and ℓ_i its length. We denote by $\boldsymbol{\kappa}$ (resp. $\boldsymbol{\kappa}^0$) the curvature vector (resp. the natural curvature vector) collecting the N scalars κ_i (resp. κ_i^0). In contrast to these geometric parameters, the physical parameters of the rod such as its lineic mass ρS or stiffness EI are assumed to remain constant along the centerline. This property is, in general, satisfied in the case of real strands.

Similarly as in [Bertails et al. 2006], the Lagrangian mechanics allows us to build a differential system for the unknown $\boldsymbol{\kappa}$,

$$\mathbb{M}(t,\boldsymbol{\kappa})\,\ddot{\boldsymbol{\kappa}} + \mathbb{K}\,(\boldsymbol{\kappa} - \boldsymbol{\kappa}^0) = \mathbf{B}(t,\boldsymbol{\kappa},\dot{\boldsymbol{\kappa}}) \qquad (1)$$

where $\mathbb{M}$ is the symmetric positive definite mass matrix with $\mathbb{M}_{ij} = \rho S \int_0^L \frac{\partial \mathbf{r}}{\partial \kappa_i}^\top \frac{\partial \mathbf{r}}{\partial \kappa_j}\,\mathrm{d}s$, $\mathbb{K}$ the constant diagonal stiffness matrix with $\mathbb{K}_{ii} = \ell_i EI$, and $\mathbf{B}$ collects the other (internal and external) forces applied onto the rod, including gravity.

5 Geometric fitting

The first step of our method automatically converts a hand-drawn curve into a G^1-smooth curve made of N circular arcs. The total number of arcs N is fixed by the user so as to control the computational cost for subsequent animation. We consider as input a set of 2d sorted points resulting from the interactive capture of the curve sketched by the user. This curve is converted into a smooth Bézier spline using a modified version of the Potrace software [Selinger 2003], where we enforce the alignment of tangents between successive Bézier patches to yield a perfectly smooth curve. The resolution of the spline is chosen high so that the conversion does not filter out any curve detail except those with very high frequencies which are considered as noise. The resulting vectorized curve is then evenly sampled into a finite list of points. Our fitting algorithm is thus initialized with a set of n ordered points together with their associated tangents and curvatures, which are computed analytically.

Our approach consists in two main passes. We first compute an optimal segmentation of the curve into exactly N pieces, based on the piecewise constant approximation of the curvature profile. This first pass provides $N+1$ gate points and tangents. We then reconstruct a smooth curve made of N circular arcs by interpolating the floating gate tangents while minimizing the error made on the gate points. We evaluate our algorithm by providing quantitative comparisons to existing approaches at the end of this section.

5.1 Segmentation of the curve into N arcs

Our approach is inspired by Pei and Horng's segmentation algorithm [Pei and Horng 1996], which makes the following observation: in order to best fit m constant pieces on the curvature profile between the first point and point i, $m-1$ constant pieces are fitted between the first point and a point $k \geq m$, and a constant piece is put between points k and i. The point k is chosen so as to minimize the approximation error of the original curvature profile, in the least-square sense. We implement this approach by designing a dynamic programming algorithm that populates a matrix $\mathbf{M}$ of size $n \times N$ in a bottom-up fashion as

$$\mathbf{M}(i,m) = min_{m \leq k < i} \left\{ \mathbf{M}(k, m-1) + E_{fit}(k,i) \right\} \qquad (2)$$

where $E_{fit}(k,l) = \sum_{i=k}^{l} |c_i - \overline{c_{kl}}|^2$, c_i are the curvatures and $\overline{c_{kl}}$ is the mean curvature $\overline{c_{kl}} = \frac{\sum_{i=k}^{l} c_i}{l-k+1}$. Note that in contrast to [McCrae and Singh 2008], our algorithm optimizes the placement of *exactly* N arcs. As a result, our formulation (2) does not require any supplementary term E_{cost} for penalizing the addition of new

arcs. However, to avoid the appearance of very small arcs that are insignificant compared to the scale and the length of the input curve, we chose to penalize short constant pieces by adding the term $E_{length}(i,m) = e^{-\frac{\ell_{im}^2}{L}}$ to our formulation (2), where ℓ_{im} is the length of the portion of curve between the points i and m, and L is the total length of the curve. In practice, we observed that adding this term yields good results in our segmentation (see section 5.3). Note that this penalty term does not require the tuning of any supplemental parameter by the user: the only parameter to be tuned is the total number N of arcs.

5.2 Reconstruction by floating tangents interpolation

We now have $N+1$ gate points $\mathbf{p}_0,\dots,\mathbf{p}_N$, along with their respective tangents $\mathbf{t}_0,\dots,\mathbf{t}_N$. The second pass of our algorithm builds N smoothly connected circular arcs by fitting *exactly one arc* between two successive gates. Note that given two successive gates i and $i+1$, it is generally impossible to interpolate both points *and* tangents with one single circular arc. The usual way to overcome this difficulty is to use *biarcs*, *i.e.*, two smoothly connected circular arcs [Bolton 1975; Nutbourne and Martin 1990; Yang and Du 1996; Drysdale et al. 2008]. In contrast, our idea is that full interpolation produces an *unnecessary over-constrained* problem.

More precisely, given an input curve, we claim that satisfying exactly the gate tangents $\mathbf{t}_0,\dots,\mathbf{t}_N$ is more important than interpolating the gate points $\mathbf{p}_0,\dots,\mathbf{p}_N$, because this is necessary for preserving the general look of the curve. Actually, the approximated curve does not need to perfectly go through the gate points, as long as (a) the general look of the initial curve is preserved and (b) the new gate points $\mathbf{p}'_0,\dots,\mathbf{p}'_N$ remain in a close neighborhood of the initial gate points $\mathbf{p}_0,\dots,\mathbf{p}_N$. This new problem statement based on the interpolation of *floating tangents* (*i.e.*, tangents that are not assigned to predefined base points) under some neighboring conditions on their new (unknown) base points gives rise to an original scheme for building an automatically G^1-smooth curve made of piecewise circular arcs. To the best of our knowledge, this algorithm was never proposed in the past, neither in computer graphics nor in computer-aided design. An obvious advantage is that, in contrast to the biarcs method, we are *not* limited to an even number N of arcs. As shown in section 5.3, our approach is also competitive in terms of reconstruction quality and computational performance, and finally offers an interesting alternative to the traditional biarcs method.

A constrained minimization problem Our goal is to minimize the distance between the original gate points $\mathbf{p}_0,\dots,\mathbf{p}_N$ and the new ones $\mathbf{p}'_0,\dots,\mathbf{p}'_N$ that should be placed so that (a) two successive new points $\mathbf{p}'_i$ and $\mathbf{p}'_{i+1}$ with their associated tangents $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$ are co-circular[1], and (b) the new gate tangents $\mathbf{t}'_0,\dots,\mathbf{t}'_N$ match the original ones $\mathbf{t}_0,\dots,\mathbf{t}_N$. This problem can be expressed as a constrained least squares minimization problem,

$$\begin{array}{ll}\min_{\{\mathbf{p}'_i\}_{i=0\dots N}} & \sum_{i=0}^{N}||\mathbf{p}'_i-\mathbf{p}_i||^2 \\ \text{subject to} & \text{(a)}\quad \mathbf{p}'_i,\mathbf{p}'_{i+1}\text{ co-circular}\quad \forall i=0\dots N-1 \\ & \text{(b)}\quad \mathbf{t}'_i=\mathbf{t}_i\quad \forall i=0\dots N.\end{array} \tag{3}$$

We now express conditions on $\mathbf{p}'_i$ so that the two above constraints are satisfied. This will transform problem (3) into the new *unconstrained* minimization problem (4).

Case of one single arc Let us first have a look at the fitting of a single arc between two gates, and introduce the following lemma,

[1]Two points $\mathbf{p}_0$ and $\mathbf{p}_1$ with their tangents $\mathbf{t}_0$ and $\mathbf{t}_1$ are *co-circular* if there exists a circle $\mathscr{C}$ passing through $\mathbf{p}_0$ and $\mathbf{p}_1$ with tangents $\mathbf{t}_0$ and $\mathbf{t}_1$.

Lemma 5.1 *Let $\mathbf{t}_0$ and $\mathbf{t}_1$ be two different unitary vectors. Then $\mathbf{p}_0$ and $\mathbf{p}_1$ are co-circular with respective (oriented) tangents $\mathbf{t}_0$ and $\mathbf{t}_1$ iff they satisfy the equation*

$$\mathbf{p}_1 = \mathbf{p}_0 + \alpha\,\mathbf{d}$$

where $\alpha\in\mathbb{R}$ and $\mathbf{d}$ is the vector built in the following way: Let $\overline{\mathbf{t}_0}$ and $\overline{\mathbf{t}_1}$ be the respective images of $\mathbf{t}_0$ and $\mathbf{t}_1$ by the rotation of angle $\frac{\pi}{2}$. Then,

$$\mathbf{d} = \overline{\mathbf{t}_1} - \overline{\mathbf{t}_0}.$$

The corresponding proof is provided in supplemental material.

General case of N arcs Given $N+1$ oriented gate tangents $\mathbf{t}_0,\dots,\mathbf{t}_N$, we can similarly build N vectors $\mathbf{d}_{k,k\in\{0\dots N-1\}}$. For each segment $k\in\{0,\dots,N-1\}$ we have $\mathbf{p}_{k+1}=\mathbf{p}_k+\alpha_k\,\mathbf{d}_k$, where $\alpha_k\in\mathbb{R}$. Starting from the initial point $\mathbf{p}'_0$, we can thus express any new gate point $\mathbf{p}'_i$ as $\mathbf{p}'_i=\mathbf{p}'_0+\sum_{k=0}^{i-1}\alpha_k\mathbf{d}_k$ where $\alpha_{k,k\in\{0\dots N-1\}}$ are real (undetermined) scalars. Our constrained minimization problem (3) is thus equivalent to solving the new unconstrained minimization problem

$$\min_{\mathbf{p}'_0,\{\alpha_k\}_{k=0\dots N-1}} \sum_{i=0}^{N}||\mathbf{p}'_0+\sum_{k=0}^{i-1}\alpha_k\,\mathbf{d}_k-\mathbf{p}_i||^2. \tag{4}$$

Note that the objective function, denoted f, is quadratic *w.r.t.* $\mathbf{p}'_0,\alpha_0,\dots,\alpha_{N-1}$, and strictly convex. It thus admits an unique minimum that is found by simply solving the $N+2$ linear equations

$$\frac{\partial f}{\partial p'_{0x}}=0\quad \frac{\partial f}{\partial p'_{0y}}=0\quad\text{and}\quad \frac{\partial f}{\partial \alpha_j}=0\quad \forall j\in\{0,\dots,N-1\}$$

where p'_{0x} and p'_{0y} are the two scalar components of $\mathbf{p}'_0$.

5.3 Evaluation of our method

Though simple, our algorithm is, to the best of our knowledge, the first one to fit G^1-connected circular arcs to a sketched curve with direct control over the number of arcs. In order to compare our work to previous methods, we adapted them so that they achieve the same goal. We propose to compare our full algorithm to the three following natural adaptations of previous approaches:

- Our segmentation method coupled with a naive adaptation of the curvature-based reconstruction by McCrae and Singh [2008], originally proposed in the case of clothoid arcs.
- Our segmentation method applied to $\frac{N}{2}$ arcs and coupled with the biarc approach for interpolating pairs of gate points and tangents with two arcs.
- The full biarc approach, directly testing the placement of $\frac{N}{2}$ biarcs between all possible pairs of points among the input data, without relying on prior segmentation.

We have implemented the construction of biarcs using the mathematical details provided in [Park 2004]. The full biarc algorithm relies on the same dynamic program as the one designed for our segmentation, where the constant curvature test between two input points is replaced with the biarc test. This kind of approach, often implemented in computer-aided design because of the high degree of precision it may offer, is however much slower than the other approaches tested here.

We have tested the 4 approaches on a panel of 10 different and representative curves depicted in figure 4. The quality comparisons

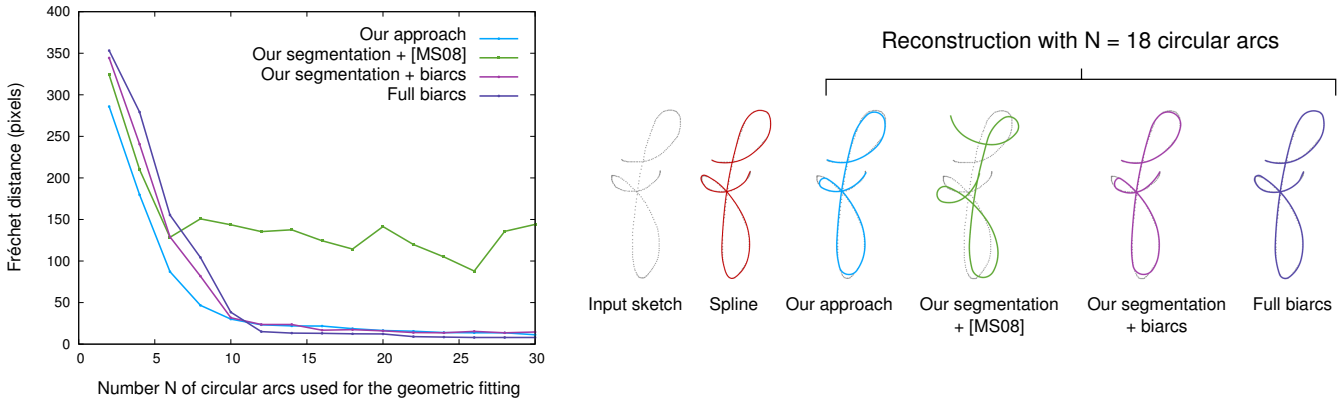


Figure 3: *Quality comparisons between the 4 reconstruction methods. Left: quantitative comparison of the Fréchet distance between the reconstruction and the input sketch (averaged on the 10 sample curves), function of the number of arcs. Right: visual comparison on a sample curve approximated with $N = 18$ arcs.*

presented in figure 3 and the performance comparisons were obtained by averaging the results computed for each individual curve. We only used *even* numbers of arcs in our tests due to the biarcs constraint. We want to highlight that the average results are actually very close to the individual ones: we did not encounter distinct behaviors from one curve to another, whatever the reconstruction approach used. An exhaustive set of comparisons in the cases when $N = 10$ and $N = 20$ arcs is provided in supplemental material.

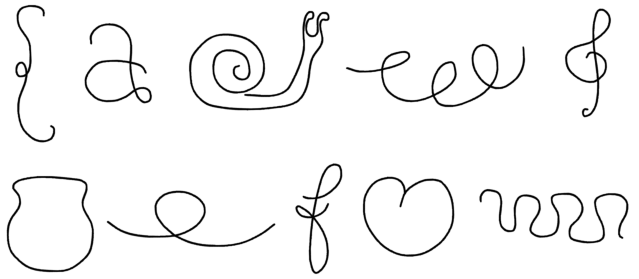


Figure 4: *The 10 sample curves (converted into splines) we used for our tests. The average number of input points is $n = 312$.*

Quality of reconstruction We measure the quality of the reconstruction by evaluating the degree of similarity between the reconstructed curve and the original one drawn by the user. This can be naturally computed using the *Fréchet distance* [Alt and Godau 1995]. In practice, we have implemented the discrete Fréchet distance algorithm introduced by Eiter and Mannila [1994]. Figure 3, left, displays the mean Fréchet distance (in pixels) function of the number N of circular arcs of the reconstruction. Note that our geometric fitting method gives the closest approximation to the original curve in the range $N = 2, \dots 14$, and keeps on giving good results for a high number of elements with a quality similar to the biarcs method coupled to our segmentation. As expected, the full biarc method yields a superior quality of reconstruction, but only for high values of N. The naive adaptation of [McCrae and Singh 2008] to circular arcs fails in capturing the initial curve geometry accurately, even in the presence of a large number of arcs. This is mainly due to the length preservation constraint.

Another interesting property of our approach compared to others is the *monotonic* decreasing of the Fréchet distance on average as N increases. The user can then reasonably expect the geometric precision to increase when increasing the resolution of the model.

Time performance Mean time performance of the different fitting algorithms based on $N = 18$ arcs was measured on a single threaded application running on an Intel Core 2 Duo CPU at 2.8 GHz. Apart from the full biarc method which is particularly slow (79 seconds on average), all approaches are fully interactive (between 23 and 42 milliseconds). Our approach as well as the adaptation of [McCrae and Singh 2008] both produce very fast piecewise arcs reconstructions (23 milliseconds).

Limitations Our geometric fitting algorithm suffers from two main limitations that correspond to extreme cases.

First, when the curve is simple with many arcs required, small arcs may be created (despite our penalty term E_{length}), possibly generating the appearance of extra loops. The reason is that points positions are not interpolated, so in case of short distances, gate points might be switched in order to satisfy the tangents interpolation constraint. See figure 5 for an illustration. This problem could be overcome by including a non-crossing constraint into the minimization process, which would result in constraining the sign of the α_k.

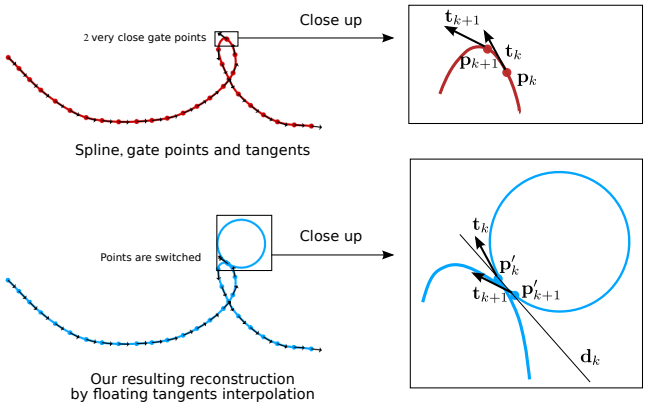


Figure 5: *One limitation of our method when a large number of arcs is required for a simple curve (28 arcs here).*

Second, when the number of arcs is clearly not sufficient for representing the input curve, the fitting can be very approximate. Imagine for example that the user chooses a single arc to match an input curve having an inflection point – *i.e.*, at least two curvatures with different sign. One way to prevent such a mismatch could be to automatically compute an estimation of the minimal number of arcs

required to guarantee a reasonable approximation – based for example on the number of inflection points of the curve. As the process is interactive, the user can alternatively manually increase the number of arcs so that he/she quickly achieves the desired resolution.

Note that apart from these degenerate cases, our method yields high-quality results that are competitive compared to previous methods, both in terms of quality and performance. The detailed set of comparison results obtained for $N = 10$ and $N = 20$ arcs is provided in supplemental material.

6 Dynamic fitting

The output of the geometric fitting is a G^1-smooth curve with clamped orientation θ_0^{fit}, made of N circular arcs with curvature κ_i^{fit} and length ℓ_i^{fit}, which closely approximates the input curve drawn by the user. Let $\boldsymbol{\kappa}^{fit}$ (resp. $\boldsymbol{\ell}^{fit}$) denote the vector collecting the N curvatures κ_i^{fit} (resp. the N element lengths ℓ_i^{fit}) resulting from the geometric fitting. Our goal is now to recover the physical parameters of a super-circle which would *exactly* generate this piecewise arcs curve as a stable equilibrium shape resulting from the counter-balance effects of elasticity and gravity. This amounts to searching for a local minimum of the potential energy E_p of the rod, evaluated at the fitted values. This problem can be mathematically formulated as

$$\text{find } (\rho S, EI, \boldsymbol{\kappa}^0) = argmin_{loc}\, E_p(\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}). \tag{5}$$

Thanks to the compact and intuitive parameterization of the super-circle model, we show in the following that this minimization problem can actually be decomposed into two simple steps:

- Find the natural curvatures vector $\boldsymbol{\kappa}^0$ yielding the equilibrium, *i.e.*, find $\boldsymbol{\kappa}^0$ such that

$$\boldsymbol{\nabla} E_p(\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}) = \mathbf{0}, \tag{6}$$

- Find the lineic mass ρS and the stiffness modulus EI yielding a stable equilibrium, *i.e.*, find ρS and EI such that

$$\boldsymbol{\nabla}^2 E_p(\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}) > \mathbf{0}. \tag{7}$$

In the next sections, we prove that these two sub-problems *always* have a solution. More precisely, the equilibrium problem (6) has a unique solution for $\boldsymbol{\kappa}^0$ which is found by trivially solving N linear scalar equations. The stable equilibrium problem (7) has an infinity of solutions and we show that we can always find a *lower bound* for the ratio $\frac{EI}{\rho S}$ that guarantees the stability of the equilibrium. Intuitively, this means that we can naturally bend and carve the rod so that, under gravity, the actual curvatures of the rod exactly match those imposed by the input geometry. Moreover, by increasing the stiffness (or reducing the mass) as much as needed, we can always reach a threshold beyond which the equilibrium becomes stable. Figure 6 demonstrates the impact of stability of the equilibrium when subsequently deforming the curve.

6.1 Potential energy of a super-circle

The potential energy of the general (space-continuous) Euler elastica under gravity reads [Bertails et al. 2005]

$$E_p = E_g + E_{el}$$

where E_{el} is the internal elastic energy of the rod,

$$E_{el} = \frac{EI}{2} \int_0^L \left(\kappa(s) - \kappa^0(s)\right)^2 \mathrm{d}s,$$

and E_g its gravitational energy (with g the constant of gravity),

$$E_g = \rho S g \int_0^L (L-s) sin(\theta(s)) \mathrm{d}s.$$

When the curvature $\kappa(s)$ is a piecewise constant function, the discrete potential energy can be analytically calculated by decomposition over each element of the rod. We refer the reader to the supplemental material for the exact expression of the discrete potential energy. The key observation is that while this is a fairly complex nonlinear function of our fitted curvatures $\boldsymbol{\kappa}^{fit}$, it appears as a simple (affine or quadratic) function of our actual unknowns ρS, EI and $\boldsymbol{\kappa}^0$. This will allow for easy derivations and intuitive results as shown in the following.

6.2 Finding an equilibrium under gravity

Our goal is to find the parameters EI, ρS and $\boldsymbol{\kappa}^0$ such that E_p is at a local minimum, implying that equation (6) is satisfied. This is actually equivalent to solving the static equation

$$\mathbb{K}(\boldsymbol{\kappa}^{fit} - \boldsymbol{\kappa}^0) = \mathbf{B}(\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}) \tag{8}$$

derived from equation (1), where the stiffness matrix $\mathbb{K}$ depends on the parameter EI and the forces vector $\mathbf{B}$ on the parameter ρS. Without any computation, a simple look at this static equation provides the answer to our problem: as EI and ρS are constant parameters along the rod, equation (8) (and thus equation (6)) admits the unique solution for $\boldsymbol{\kappa}^0$,

$$\boldsymbol{\kappa}^0 = \boldsymbol{\kappa}^{fit} - \mathbb{K}^{-1}\mathbf{B}(\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}) \tag{9}$$

while EI and ρS can freely span the entire positive real space.

Note that the expression for $\boldsymbol{\kappa}^0$ which satisfies the equilibrium depends on the EI and ρS parameters. It should thus be re-evaluated each time one of these parameters is modified (see our full algorithm in Figure 7).

6.3 Finding a stable equilibrium under gravity

Evaluating the stability of the equilibrium requires the computation of the Hessian matrix $\nabla^2 E_p$: the equilibrium will be stable if $\nabla^2 E_p$ is a positive-definite matrix, *i.e.*, characterized by strictly positive eigenvalues[2].

Expression of the Hessian The analytical derivation of the Hessian is provided in supplemental material. The key is to observe that $\nabla^2 E_p$ is a symmetric matrix that *only* depends on the ρS and EI parameters (and not on $\boldsymbol{\kappa}^0$). Furthermore, this dependency is *linear*, and the EI parameter is located on the *diagonal terms* of the matrix. More precisely, $\nabla^2 E_p$ can be expressed as

$$\nabla^2 E_p = EI\,\mathbb{D} + \rho S\,\mathbb{S} \tag{10}$$

where $\mathbb{D}$ is a diagonal matrix with $\mathbb{D}_{ii} = \ell_i, \forall i = 1 \ldots N$, and $\mathbb{S}$ a dense symmetric matrix. Note that the two matrices $\mathbb{D}$ and $\mathbb{S}$ are *independent* of the parameters EI, ρS, and $\boldsymbol{\kappa}^0$.

[2] This condition is actually sufficient but not necessary for having a stable equilibrium. When some eigenvalues of the Hessian are zero, one should theoretically have a look at the sign of the higher order derivatives of E_p. In practice however, we never encountered such a case and thus limited our stability study to the second order.

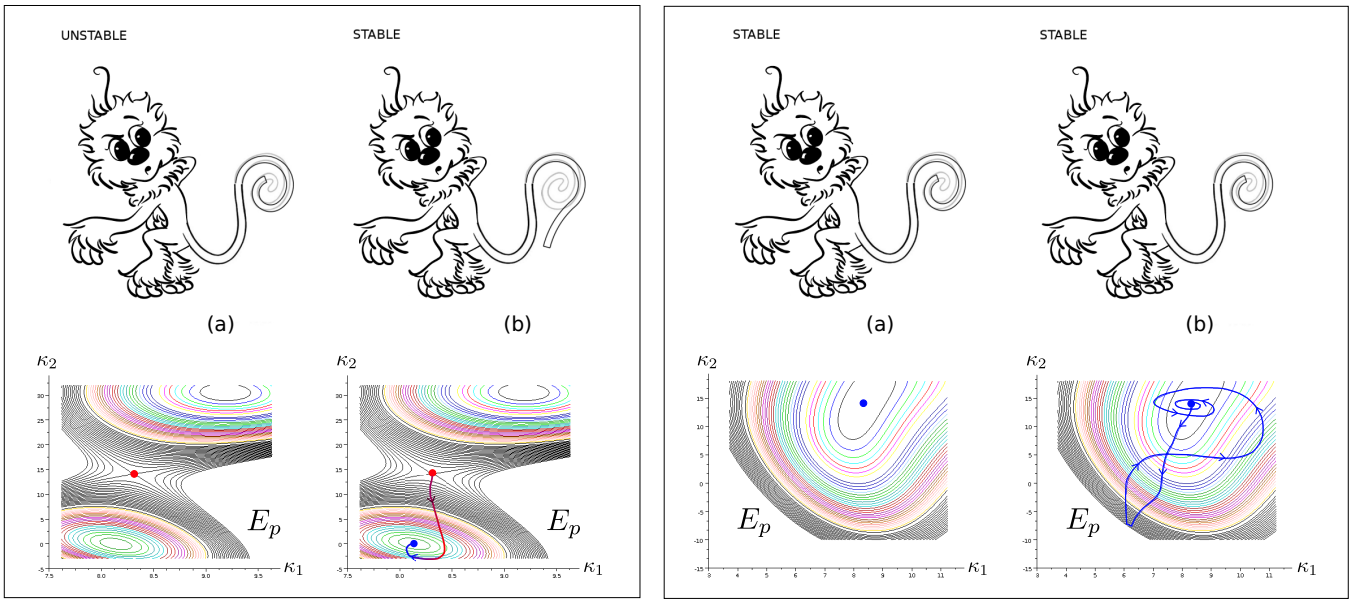


Figure 6: *Impact of the equilibrium stability on the example of a character's tail approximated by $N = 2$ circular arcs. After matching the input curve as an equilibrium position of the super-circle model (a), the experiment consists in applying vanishing forces to the rod (here, the user pulls the tail to the right, and then lets it go) and observing the new equilibrium state (b). Our stability algorithm was only applied in the case depicted on the right, where we indeed observe a stable rest position as opposed to the left part of the figure. The top row shows the 2d configuration of the character's tail. In the bottom row we have plotted, with the help of Scilab (http://www.scilab.org), the contour lines of the potential energy as well as the trajectory followed by the two curvatures κ_1 and κ_2 during the experiment. As expected, the rod comes back to its original rest shape in the case when the equilibrium is stable (potential well, in blue), in contrast to the unstable equilibrium case where the new rod configuration (necessary a local minimum) may greatly differ from the original one (here a saddle point, in red).*

A sufficient condition for stability Our goal is to find a condition on the remaining parameters EI and ρS so that the eigenvalues of $\nabla^2 E_p$ are strictly positive. The study of eigenvalues is a difficult research field which is still active in mathematics. Results related to eigenvalues of a sum of real symmetric matrices have been obtained a few decades ago only [Fulton 2000]. We use such a result to find a sufficient condition on $\frac{EI}{\rho S}$ so that the Hessian $\nabla^2 E_p$ is positive-definite.

As $\mathbb{S}$ and $\nabla^2 E_p$ are real symmetric matrices, they are diagonalizable. Let $\lambda_1, \ldots, \lambda_N$ (resp. $\mu_1, \ldots, \mu_N$) be the eigenvalues of $\mathbb{S}$ (resp. of $\nabla^2 E_p$) sorted in decreasing order. Similarly, let $\ell'_1, \ldots, \ell'_N$ be the sorted eigenvalues (diagonal terms) of $\mathbb{D}$. Applying Horn's theorem [Fulton 2000], we have

$$\forall i, j, k \text{ such that } i + j = N + k, \qquad \mu_k \geq EI\,\ell'_j + \rho S\,\lambda_i,$$

which, in the special case when $k = N$ and $i = j = N$, reads

$$\mu_N \geq EI\,\ell'_N + \rho S\,\lambda_N \tag{11}$$

with $\mu_N = min_j\{\mu_j\}$, $\ell'_N = min_j\{\ell_j\}$, and $\lambda_N = min_j\{\lambda_j\}$.

Equation (11) provides a lower bound for the smallest eigenvalue of $\nabla^2 E_p$. It thus yields a sufficient condition on $\frac{EI}{\rho S}$ for guaranteeing that *all* the eigenvalues μ_i of $\nabla^2 E_p$ are strictly positive,

$$\frac{EI}{\rho S} > -\frac{\lambda_N}{\ell'_N}. \tag{12}$$

We remind the reader that λ_N and ℓ'_N are frozen once the curve rest shape has been provided by the user (*i.e.*, they are independent of our parameters ρS, EI and $\boldsymbol{\kappa}^0$). According to equation (12), by either *increasing* EI or *decreasing* ρS, the stability of the rest shape under gravity can thus *always* be guaranteed, whatever the input curve is. More precisely, we can define the thresholds

$$EI_{min} = -\rho S\,\frac{\lambda_N}{\ell'_N} + \varepsilon_{EI} \qquad \text{and} \qquad \rho S_{max} = -EI\,\frac{\ell'_N}{\lambda_N} - \varepsilon_{\rho S}$$

for the parameters EI and ρS, where ε_{EI} and $\varepsilon_{\rho S}$ are arbitrary small positive values. The equilibrium will be guaranteed to be stable if $EI = EI_{min}$ or $\rho S = \rho S_{max}$. The choice for playing on one or the other parameter is left to the user for letting him/her precisely control the stiffness or the mass of the rod.

How to increase stability The potential well is all the more steep as the eigenvalues μ_i are large. This means that the level of stability – or the attraction power of the equilibrium – can still be improved by increasing EI (resp. decreasing ρS) beyond the threshold EI_{min} (resp. ρS_{max}). In practice, this can be useful for an animator who would like to have the rod recover its initial configuration even after being largely displaced and deformed. In the video game application (see section 7), we have assigned EI to $1.5\,EI_{min}$ so that the rope is guaranteed to always come back to its initial rest shape, whatever the amplitude of deformations.

Time performance Computing a stable configuration for a given curve consists in two steps: first, evaluating the lower bound of equation (12) and initializing EI (or ρS) accordingly, and second, solving the diagonal system (9) of size N. For curves made of tens of circular arcs, these computations are instantaneous. When converting the piecewise reconstructions of our 10 sample curves (depicted in figure 4) into stable rest shapes, we have measured a mean computational time ranging between 0.1 milliseconds (for N = 10 arcs) and 0.5 milliseconds (for N = 30 arcs).

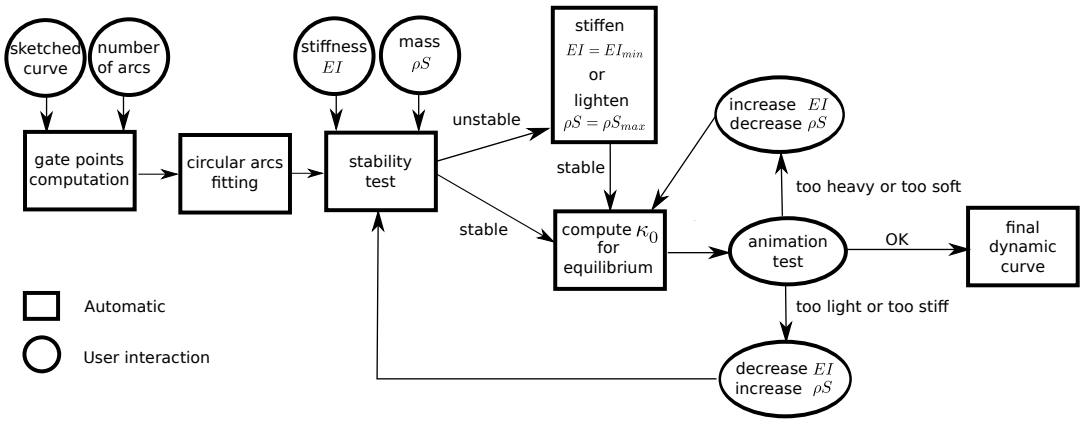


Figure 7: *Our full algorithm, from the user's viewpoint.*

7 Results

Figure 7 summarizes the algorithm provided to the user for initializing an animation with the guarantee of starting with a stable configuration. Based on this pipeline we present two applications: a 2d animation creation tool and a simple video game.

Animation design Our first application aims at producing secondary motions in 2d animation. Figure 8 and the accompanying video show an example where the user sketches some hair strands attached to a head modeled as a simple textured quad. The sketched curves are automatically converted into super-circles with a given number of arcs (we typically chose 8 or 10 arcs per curve). When the head moves, either using a predefined motion or via user interaction, the hair follows the head motion, giving rise to natural secondary motions. To obtain our final result we simply render each curve using a black colored triangle strip with a width linearly decreasing along the curve.

Such applications show how intuitive it is for an artist to design a physics-based animation from an input geometry. The stable equilibrium computation guarantees that the hair strands will recover their initial shape most of the time. Of course, very large head motions may lead to another equilibrium shape, as in the real world.

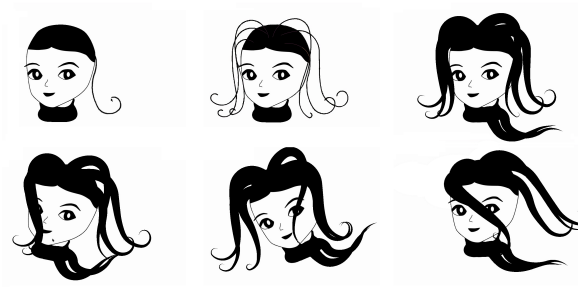


Figure 8: *2d animation application. The user sketches some hair strands and a scarf. Once they have been converted into dynamic strands, they naturally follow the head motion and recover their initial shape when the motion ends.*

Physically-based Video Game Our second application is a small video game whose goal is to make a character jump between two blocks using a dynamic rope. We show an example in figure 9 and in the accompanying video. The user first draws a curve between a given anchor point and the character. Then, he/she moves the character horizontally (using arrow keys). When ready to perform the jump, he/she lets the character be pulled by the rope (by pressing space bar) until making the character drop the rope (space bar again).

This application demonstrates again the need for a stable equilibrium under gravity. This property is typically used by the game player to pull the character by letting the rope go back to its original shape. It also shows an example where the geometry is totally part of the curve behavior and therefore of the game design. As shown in the video, designing a straight curve causes a failure because it may be too short and may also lack elasticity. The user has to use well placed loops to compensate for the inextensibility of the rod. After several tests and trials, various funny and intuitive solutions are possible to win the game.

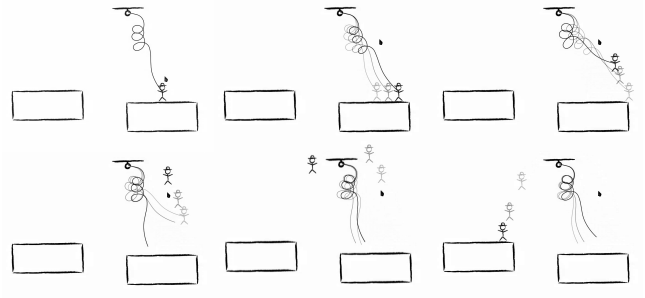


Figure 9: *Video game application. By drawing a curve that is automatically converted into a dynamic rope, the user tries to make the character jump from the right block to the left one.*

Limitations and Future Work Our approach is currently limited to 2d smooth open curves clamped at one end, subject to a known external force such as gravity. Based on the recursive algorithm for a super-helix [Bertails 2009], we show in the supplemental video a preliminary extension to tree-like structures. In the short term we plan to extend our dynamic fitting to account for various constraints applied to the rod model (such as multiple anchor points) where the constraint forces would be part of the unknowns in the stability problem. In the longer term we would like to extend our approach to the handling of sharp corners as well as closed contours, which have a lot of applications in 2d animation. Our space control approach could also be interestingly combined to motion control techniques such as the one described in [Barbič and Popović 2008], in order to produce a full key-framing process. Finally, extending our method

to the 3d coupling between a skew curve and a rod model with twist would open the way for a large number of applications in reverse engineering. Note that our dynamic fitting can be straightforwardly extended to super-helices. Actually the only difficult part would be the geometric fitting into piecewise helices.

8 Conclusion

We have presented a new method for automatically fitting a sketched curve into the stable rest shape of a dynamic rod under gravity. Our approach relies on two original algorithms, one dedicated to the *geometric* fitting with precise control of the resolution, and the other one to the *dynamic* fitting with precise control of the stability. We have demonstrated the advantages of our approach on animation design and physically-based video game. We believe that such a method opens the way for new intuitive interfaces for coupling input geometry and dynamics.

Acknowledgments

We would like to thank Laurence Boissieux for her artistic contribution to the paper, Basile Audoly for sharing with us the original code on 2d super-helices, and the anonymous reviewers for their fruitful comments. Alexandre Derouet-Jourdan is supported by a grant from the École Normale Supérieure of Lyon.

References

ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'00 conference)*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, ACM, 157–164.

ALT, H., AND GODAU, M. 1995. Computing the fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl. 5*, 75–91.

BARAN, I., LEHTINEN, J., AND POPOVIĆ, J. 2010. Sketching clothoid splines using shortest paths. *Computer Graphics Forum (Proceedings of Eurographics'10)*. to appear.

BARBIČ, J., AND POPOVIĆ, J. 2008. Real-time control of physically based simulations using gentle forces. *ACM Trans. on Graphics (SIGGRAPH Asia 2008) 27*, 5, 163:1–163:10.

BARZEL, R. 1997. Faking dynamics of ropes and springs. *IEEE Comput. Graph. Appl. 17*, 3, 31–39.

BAXTER, W., BARLA, P., AND ANJYO, K.-I. 2009. N-way morphing for 2d animation. *Computer Animation and Virtual Worlds (Proceedings of CASA) 20*, 2.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'08 conference) 27*, 3, 1–12.

BERTAILS, F., AUDOLY, B., QUERLEUX, B., LEROY, F., LÉVÊQUE, J.-L., AND CANI, M.-P. 2005. Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics'05 (short papers)*, J. Dingliana and F. Ganovelli, Eds., Eurographics.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'06 conference) 25*, 3, 1180–1187.

BERTAILS, F. 2009. Linear time super-helices. *Computer Graphics Forum (Proceedings of Eurographics'09) 28*, 2 (apr).

BOLTON, K. 1975. Biarc curves. *Journal of Computer Aided Design 7*, 89–92.

DRYSDALE, R. S., ROTE, G., AND STURM, A. 2008. Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. *Computational Geometry: Theory and Applications 41*, 1-2, 31–47.

EITER, T., AND MANNILA, H. 1994. Computing discrete fréchet distance. Tech. rep., Technische Universitat Wien.

FEATHERSTONE, R. 1983. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research 2*, 1, 13–30.

FULTON, W. 2000. Eigenvalues, invariant factors, highest weights, and schubert calculus. *Bull. Amer. Math. Soc 37*, 209–249.

HADAP, S. 2006. Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'06)*, ACM-EG SCA, 91–100.

HORNG, J.-H. 2003. An adaptive smoothing approach for fitting digital planar curves with line segments and circular arcs. *Pattern Recogn. Lett. 24*, 1-3, 565–577.

MCCRAE, J., AND SINGH, K. 2008. Sketching piecewise clothoid curves. *Computers & Graphics 33*, 4, 452–461.

NUTBOURNE, A., AND MARTIN, R. 1990. *Differential Geometry applied fo Curve and Surface Design*. Ellis Horwood.

PARK, H. 2004. Optimal single biarc fitting and its applications. *Computer-Aided Design and Applications 1*, 1-4, 187–195.

PEI, S.-C., AND HORNG, J.-H. 1996. Optimum approximation of digital planar curves using circular arcs. *Pattern Recognition 29*, 3, 383–388.

SAFONOVA, A., AND ROSSIGNAC, J. 2003. Compressed piecewise-circular approximations of 3d curves. *Computer-Aided Design 35*, 6, 533–547.

SELINGER, P., 2003. Potrace: a polygon-based tracing algorithm. http://potrace.sourceforge.net/.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'08 conference) 27*, 3, 1–11.

SPILLMANN, J., AND TESCHNER, M. 2007. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'07)*, ACM-EG SCA, 63–72.

SPILLMANN, J., AND TESCHNER, M. 2008. An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum (Proceedings of Eurographics'08) 27*, 2.

THEETTEN, A., GRISONI, L., ANDRIOT, C., AND BARSKY, B. 2008. Geometrically exact splines. *Journal of Computer Aided Design 40*, 1, 35–48.

WITHER, J., BERTAILS, F., AND CANI, M.-P. 2007. Realistic hair from a sketch. In *International Conference on Shape Modeling and Applications, SMI 2007, June, 2007*, IEEE, Lyon, France, IEEE, 33–42.

YANG, S.-N., AND DU, W.-C. 1996. Numerical methods for approximating digitized curves by piecewise circular arcs. *J. Comput. Appl. Math. 66*, 1-2, 557–569.