

Interactive generation of calligraphic trajectories from Gaussian mixtures

Daniel Berio, Frederic Fol Leymarie, Sylvain Calinon

Abstract The chapter presents an approach for the interactive definition of curves and motion paths based on Gaussian mixture model (GMM) and optimal control. The input of our method is a mixture of multivariate Gaussians defined by the user, whose centers define a sparse sequence of keypoints, and whose covariances define the precision required to pass through these keypoints. The output is a dynamical system generating curves that are natural looking and reflect the kinematics of a movement, similar to that produced by human drawing or writing. In particular, the stochastic nature of the GMM combined with optimal control is exploited to generate paths with natural variations, which are defined by the user within a simple interactive interface. Several properties of the Gaussian mixture are exploited in this application. First, there is a direct link between multivariate Gaussian distributions and optimal control formulations based on quadratic objective functions (linear quadratic tracking), which is exploited to extend the GMM representation to a controller. We then exploit the option of tying the covariances in the GMM to modulate the style of the calligraphic trajectories. The approach is tested to generate curves and traces that are geometrically and dynamically similar to the ones that can be seen in art forms such as calligraphy or graffiti.

1 Introduction

The hand-drawn curves that can be observed in art forms such as calligraphy [32] and graffiti [8] are often the result of skillful and expressive movements that require years to master. Even after practice, the same trace executed twice will always be different due to motor variability. Mimicking this type of curves and variability with

Daniel Berio, Frederic Fol Leymarie
Goldsmiths, University of London, e-mail: d.berio@gold.ac.uk, ffl@gold.ac.uk

Sylvain Calinon
Idiap Research Institute, e-mail: sylvain.calinon@idiap.ch

This is an author's version of the chapter appearing on "Mixture models for the analysis, edition, and synthesis of continuous time series". The final publication is available at Springer in the book "Mixture Models and Applications" (2019), pp. 23-38, edited by Bouguila, N. and Fan, W.

conventional geometric computer aided design (GCAD) methods can be difficult. These methods typically describe a curve through the concatenation of piecewise polynomials, which interpolate or approximate the vertices of a control polygon defined by a user. This approach is well suited for geometric design applications. However, the manual positioning of control points can become unintuitive and overly complex when the task at hand requires mimicking the curvilinear patterns that would be produced by the movements of an experienced artist’s hand. To this end, we propose a *movement centric* approach to curve design, in which a curve is defined through the synthesis of a movement underlying its production rather than only considering its static geometric trace.

In this chapter we demonstrate how tools from statistics and optimal control, together with insights from computational motor control, can be combined into a curve generation method that produces synthetic traces that are visually and kinematically similar to the ones made by a human when drawing or writing. The *input* of our method is a *Gaussian mixture model* (GMM)¹ describing a spatial distribution. The output of our method is a distribution of smooth trajectories, with variations and kinematics that are similar to the ones that typically characterize human hand motions. We generate smooth trajectories by forcing a dynamical system to track the centers of each GMM component with a precision determined by the respective covariances. The trajectory evolution is determined by an optimization with a quadratic objective, which is formulated as a trade-off between tracking accuracy and control effort. The latter is expressed as the square magnitude of a n^{th} order derivative of position, such as jerk (3^{rd}) or snap (4^{th}), which results in smooth trajectories that are consistent with known principles from computational motor control [18, 41]. Accompanying source codes for the chapter are available at <http://doc.gold.ac.uk/autograff/>.

2 Background

The proposed approach is informed by a number of observations and ideas from the domain of computational motor control. Target-directed hand movements are characterized by an archetypal “bell” shaped velocity profile [34, 36, 17]. A number of mathematical models of handwriting movements describe trajectories as the time superposition of multiple target-directed sub-movements [37, 16], where each sub-movement is in turn characterized by a bell-shaped velocity profile. The speed and curvature of human hand movements tend to show an inverse relation [21, 15] with this relation taking the form of a power law for certain types of movements [27, 44]. The duration of hand movements tends to be approximately invariant to scale, a principle that is also known as *isochrony*. Also the duration of sub-movements tends to be approximately equal, a principle that is commonly referred to as *local isochrony* [25], and which is consistent with the hypothesis of central pattern generators [14]. Human hand movements are smooth and appear to obey optimality principles, which

¹ We refer the reader to the chapter by O.E. Parsons in this same book for an introduction and in-depth description of GMMs and relevant estimation methods.

can be well modeled as the minimization of an objective function [19]. Popular models express this objective as the minimization of the square-magnitude of higher order positional derivatives [18, 12, 42, 33] or torque [43]. Human movements show inherent variability [6], which tends to increase in parts of the movement that are not critical to the required precision of a task. Todorov and Jordan [41] propose the framework of optimal feedback control, and suggest that deviations from an average (smooth) trajectory are corrected only when they interfere with the required task precision. Our method allows to model this principle by explicitly defining the required precision of trajectory segments with full covariance matrices.

Egerstedt and Martin [13] discuss the equivalence between several forms of splines and control theoretic formulations of dynamical systems. The authors show that smoothing splines correspond to the output of a controller found by minimizing a quadratic cost function similar to the one used in our method. In a related line of work, Fujioka *et al.* [22] optimize the placement of B-spline [10] control points in order to mimic smoothing effects observable in Japanese calligraphy. With our method we extend these principles to a more generic case, in which movement precision as well as coordination patterns are encoded as full covariance matrices, and where the output of the method is a distribution rather than a single trajectory.

In conventional computer graphics applications, hand-drawn curves are usually specified interactively through a sketch-based interface. A user traces a curve with a mouse, trackpad or tablet. The trace is then processed in order to avoid digitization artefacts and hesitations, with a procedure commonly referred to as curve “neatening” or “fairing” [31, 2, 40, 30]. However, the output of these methods is usually a piecewise polynomial curve set with several control points, and this makes it difficult to later edit or vary the overall trace.

Non photorealistic animation and rendering (NPAR) is the subfield of computer graphics aimed at the simulation of artistic techniques/styles and at clarity of representation [26]. A few methods from this domain also target the generation of curves through the simulation of a hand movement. Haeberli [23] uses a mass-spring system to generate calligraphic curves from input mouse movements. House and Singh [24] use a proportional-integral-derivative (PID) controller to generate sketch-based stylizations of 3D models. AlMeraj *et al.* [1] mimic the quality of hand-drawn pencil lines with a model of human movements that minimizes changes in acceleration [19]. The method we describe in this chapter allows to achieve similar artistic rendering effects or to generate curves that are similar to the ones produced by a sketch-based interface. At the same time, we provide a user-friendly interface to input a sparse sequence of key-points, bearing similarities to the interfaces used in conventional GCAD methods.

3 Trajectory Generation

The input to our method is a GMM with M multivariate components $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ defined in a Cartesian space of dimension D . The output is a distribution of smooth

motions $\mathcal{N}(\mathbf{x}, \Sigma^x)$, where each motion tracks the centers μ_i of the input with a precision defined by the corresponding covariances Σ_i . Considering a sequence of centers of the mixture components gives a series of *key-points* (μ_1, \dots, μ_M) , results in a descriptor that is similar to the control polygon used in conventional curve generation methods such as Bézier curves or splines. At the same time, the covariance structure of the GMM provides explicit control over the variability and smoothness of the trajectory in the neighborhood of each key-point, together with local or global control of the curvilinear evolution of the trajectory (Fig. 1).

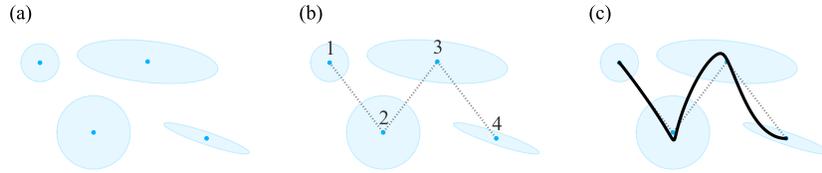


Fig. 1 The trajectory generation method in a nutshell. An input GMM (a) is considered as a sequence (b). The ordered components are then used to guide the evolution of a dynamical system (c).

Trajectories are generated by optimizing the evolution of a dynamical system that tracks each GMM component sequentially for a given amount of time. A decrease in the variance of a component corresponds to an increased precision requirement and thus forces the trajectory to pass near the component center (Fig. 2a). A sufficiently low variance then produces an interpolatory behavior. An increase in the variance corresponds with a lower precision requirement, and thus produces a smoothing effect that is similar to the one achieved with approximating splines (Fig. 2b). However, the use of full covariances allows more complex spatial constraints to be captured, such as forcing a movement to follow a given direction or to pass through a narrow region of space (Fig. 2c). The resulting trajectories are smooth and have kinematics that are similar to the ones that would be seen in a movement made by a drawing hand, with desirable features such as bell-shaped speed profiles and an inverse relation between speed and curvature.

3.1 Dynamical system

The model for our trajectory generation mechanism is a discrete linear time-invariant system of order n defined with the state equation:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \quad (1)$$

where at each time step t the state,

$$\mathbf{x}_t = \left[\mathbf{x}^T, \dot{\mathbf{x}}^T, \ddot{\mathbf{x}}^T, \dots, \mathbf{x}^{(n-1)T} \right]^T, \quad (2)$$

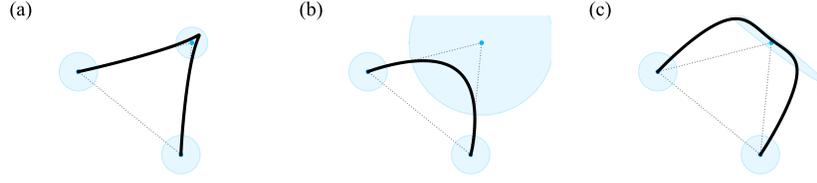


Fig. 2 Variations of a trajectory by manipulating one covariance matrix. **(a)** Using an isotropic covariance with low variance (high precision). **(b)** An increase in variance produces a smoothing effect. **(c)** A full covariance can be used to force the trajectory to remain in a restricted (here nearly flat) region of space.

concatenates the position and its derivatives up to order $n - 1$. The matrices A and B describe the time invariant response of the system to an input command u_t . For the examples presented here, we utilize a chain of n integrators commanded by its n -th order derivatives. The system matrices for the continuous version of this system are then given by:

$$A^c = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \quad B^c = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \quad (3)$$

with I a $D \times D$ identity matrix. The discrete time versions of the system matrices can be computed with a forward Euler discretization:

$$A = \Delta t A^c + I \quad \text{and} \quad B = \Delta t B^c, \quad (4)$$

where Δt is the duration of one time step, or with higher order approximation methods such as zero order hold (ZOH).

The positions along the trajectory are then given by:

$$y_t = C x_t, \quad \text{where} \quad C = [I, \mathbf{0}, \dots, \mathbf{0}, \mathbf{0}]. \quad (5)$$

From a control perspective, the sensor matrix C determines what elements of the state are observed in a feedback system. For our use case of curve generation, this formulation of C limits the parameters of our method to the position components of the state, which greatly simplifies the user interaction with the method.

3.2 Optimization objective

We generate a trajectory of T time steps by computing an optimal controller that minimizes a quadratic cost, which penalizes a trade-off between deviations from

a reference state sequence $\{\bar{\mathbf{x}}_t\}_{t=1}^T$ (*tracking cost*) and the magnitude of a control command sequence $\{\mathbf{u}_t\}_{t=1}^{T-1}$ (*control cost*). The optimization objective is expressed with the cost:

$$J = \sum_{t=1}^T (\bar{\mathbf{x}}_t - \mathbf{x}_t)^\top \mathbf{Q}_t (\bar{\mathbf{x}}_t - \mathbf{x}_t) + \sum_{t=1}^{T-1} \mathbf{u}_t^\top \mathbf{R}_t \mathbf{u}_t \quad , \quad (6)$$

subject to the constraint of the linear system defined in Eqn. 1, with \mathbf{Q}_t and \mathbf{R}_t being positive semi-definite weight matrices that determine the tracking and control penalties for each time step. The linear constraint guarantees that the output of the method is a trajectory that has continuous derivatives up to order $n - 1$.

The combination of a linear system with this type of optimization objective is commonly used in process control and robotics applications, where it is known as discrete linear quadratic tracking (LQT) and corresponds to the quadratic cost case of model predictive control (MPC) [45]. This results in a standard optimization problem that can be solved iteratively or in batch form and produces an optimal controller or control command sequence. In typical control settings, the optimization is performed iteratively over a time horizon of observations, and is thus commonly known as receding horizon control. However, for the intended use case of curve design, we can apply the optimization to the full duration of the desired trajectory. With the appropriate formulation of the reference, this results in a flexible curve generation method that can be used similarly to the more conventional ones.

3.3 Tracking formulation

We formulate the reference state and weights for the optimization objective, by pairing each input Gaussian with an activation function:

$$h_i(t) = \frac{\phi_i(t)}{\sum_{j=1}^m \phi_j(t) + \epsilon} \quad , \quad \text{with} \quad \phi_i(t) = \exp\left(-\frac{(t - \tau_i)^2}{2\sigma^2}\right) \quad , \quad (7)$$

where τ_i defines the *passage time* for the state, σ is a global parameter which defines the *time interval* covered by each state and ϵ is an arbitrarily small value that avoids divisions by 0.

With an assumption of local isochrony, we define the passage times for each state at equidistant time steps with: $\tau_{i+1} - \tau_i = T/(M - 1)$, and with $\tau_1 = 1$ and $\tau_M = T$.

The reference states and weights are then generated by assigning to each time step the state for which $h_i(t) > 0.5$ (Fig. 3, second row) with:

$$\bar{\mathbf{x}}_i = \mathbf{C}^\top \boldsymbol{\mu}_i \quad \text{and} \quad \mathbf{Q}_i = \mathbf{C}^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{C} \quad . \quad (8)$$

With this formulation the derivatives of the trajectory are fully determined by the optimization procedure, which is expressed by setting the corresponding precision

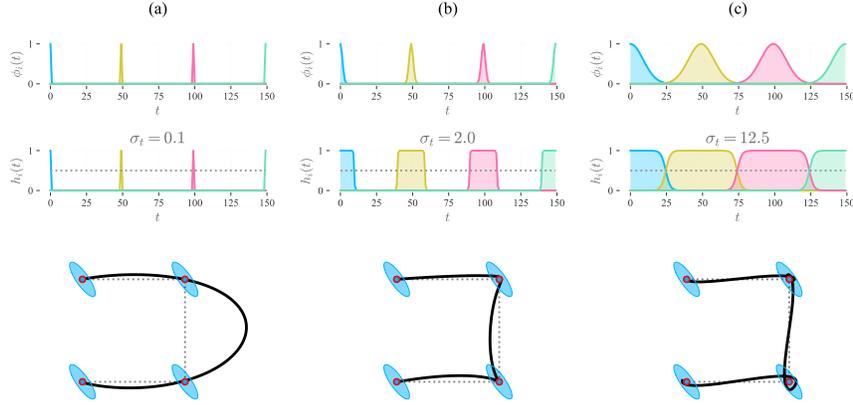


Fig. 3 Effect of three different activation sequences with the same set of Gaussians.

terms \mathbf{Q}_t to zero. Intuitively, a zero entry in \mathbf{Q}_t means that the optimization has no precision requirements for the corresponding state entry and thus is free to enforce the smoothness requirement expressed in the second term of the cost function. In typical applications, the tracking weights \mathbf{Q}_t are defined as diagonal matrices. This corresponds to a penalty in terms of the Euclidean distance to a reference state. In our stochastic formulation, the weights are expressed as full precision matrices, which correspond to a penalty in terms of the Mahalanobis distance to the reference state. When it is desirable to force the movement to a full stop, this can be done by setting $\mathbf{Q}_N = \mathbf{I}$ and all the derivative terms in $\bar{\mathbf{x}}_N$ to zero.

Increasing the value of σ increases the time interval covered by a state, with $\sigma = T/(4(M - 1))$ resulting in a stepwise reference that fully covers the time steps of the trajectory (Fig. 3c). This increases the influence of the GMM covariances on the resulting trajectory, and allows a user to specify curvilinear trends and variability for longer segments of the trajectory. As the parameter σ tends to zero, $\phi_i(t)$ will converge to a delta function (Fig. 3a), which will result in \mathbf{Q}_t being non-zero only in correspondence with each passage time τ_i . This will result in smoother trajectories that interpolate the key-points. In general, a lower time interval will result in sparser tracking cost in the objective. This increases the influence of the control cost and potentially facilitates the addition of objectives and constraints to the optimization.

3.4 Stochastic solution

The optimal trajectory can be retrieved iteratively using (1) dynamic programming [7, 4], or (2) in a batch form by solving a large ridge regression problem. Here we describe the latter, which results in a more compact solution and additional flexibility, such as a straightforward probabilistic interpretation of the result. To compute the

solution, we exploit the time invariance of the system and express all future states as a function of the initial state \bar{x}_1 with:

$$\mathbf{x} = \mathbf{S}^x \bar{x}_1 + \mathbf{S}^u \mathbf{u}, \quad (9)$$

where

$$\mathbf{S}^x = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix} \quad \text{and} \quad \mathbf{S}^u = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix}. \quad (10)$$

We then express the objective (6) in matrix form as:

$$J = (\bar{\mathbf{x}} - \mathbf{x})^\top \mathbf{Q} (\bar{\mathbf{x}} - \mathbf{x}) + \mathbf{u}^\top \mathbf{R} \mathbf{u}, \quad (11)$$

where \mathbf{Q} and \mathbf{R} are large block matrices with \mathbf{Q}_t and \mathbf{R}_t along their block diagonals, while $\bar{\mathbf{x}}$, \mathbf{x} and \mathbf{u} are column vectors representing the reference, state and control commands, this for each time step. Substituting (9) into (11), differentiating with respect to \mathbf{u} , and setting to zero result in a ridge regression solution of the form:

$$\mathbf{u} = \underbrace{((\mathbf{S}^u)^\top \mathbf{Q} \mathbf{S}^u + \mathbf{R})^{-1}}_{\boldsymbol{\Sigma}^u} (\mathbf{S}^u)^\top \mathbf{Q} (\bar{\mathbf{x}} - \mathbf{S}^x \bar{x}_1), \quad (12)$$

which is then substituted back into (9) to generate a trajectory.

From Eqn. 12 we can see that \mathbf{R} effectively acts as a Tikhonov regularization term in the least squares solution, resulting in a global smoothing effect on the generated trajectory.

From a probabilistic perspective, \mathbf{R} corresponds to a Gaussian prior on the deviations of the control commands from $\mathbf{0}$. The minimization of Eqn. 11 can then be interpreted as the product of two Gaussians:

$$\mathcal{N}(\mathbf{u}, \boldsymbol{\Sigma}^u) \propto \mathcal{N}\left((\mathbf{S}^u)^\dagger (\bar{\mathbf{x}} - \mathbf{S}^x \bar{x}_1), (\mathbf{S}^u)^\top \mathbf{Q} \mathbf{S}^u\right) \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad (13)$$

describing a distribution of control commands with center \mathbf{u} and covariance $\boldsymbol{\Sigma}^u$. By using the linear relation (9) the distribution in control space can also be interpreted as a *trajectory distribution*:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\Sigma}^x) \quad \text{with} \quad \boldsymbol{\Sigma}^x = \mathbf{S}^u \boldsymbol{\Sigma}^u (\mathbf{S}^u)^\top. \quad (14)$$

This formulation results in a generative model of trajectories, which can be used to generate variations that are similar to the ones that would be seen in multiple instances of human writing or drawing (Fig. 4). Because of its lower dimensionality, it is preferable to generate variations at the control level, which can be done by exploiting the eigendecomposition:

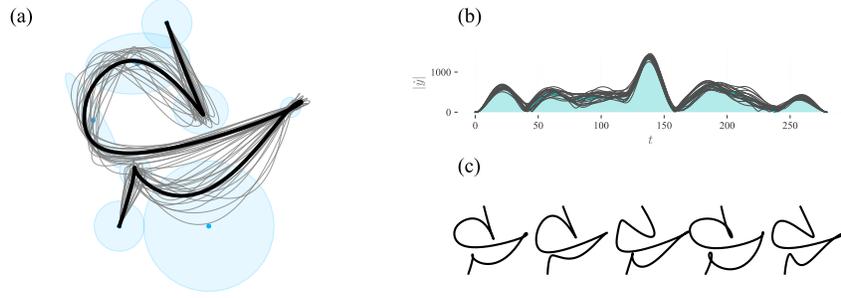


Fig. 4 Stochastic sampling. **(a)** GMM with corresponding trajectory (dark thick race) overlaid together with samples from the trajectory distribution (light gray traces). **(b)** Corresponding sampled speed profiles. **(c)** A few samples selected from the trajectory distribution.

$$\Sigma^u = V^u D^u (V^u)^\top, \quad (15)$$

where V^u denotes a matrix with all eigenvectors along the columns and D^u denotes a matrix with the corresponding eigenvalues along its diagonal. We can then generate samples around the average commands sequence u with:

$$u' \sim u + V^u (D^u)^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \sigma^u I), \quad (16)$$

where σ^u is a user-defined parameter to select the desired sample variation. The resulting trajectories can then easily be retrieved by fitting in the selected samples u' back into Eqn. 9, see Fig. 4 for the results.

3.5 Periodic motions

In order to generate periodic motions (Fig. 5), we can reformulate the LQT objective with the addition of an equality constraint on the initial and final states of the trajectory. This can be formulated with the linear relation:

$$Kx = K(S^x \bar{x}_1 + S^u u) = \mathbf{0}, \quad (17)$$

with K a matrix with zero blocks for each time step apart for the ones corresponding to the states desired to be equal. Adding this constraint to Eqn. 11 results in the Lagrangian:

$$\mathcal{L}(u, \lambda) = J + \lambda^\top Kx. \quad (18)$$

Differentiating for u and the Lagrange multipliers λ and then equating to $\mathbf{0}$ results in the following constrained solution, in matrix form:

$$\begin{bmatrix} u \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} (\Sigma^u)^{-1} (S^u)^\top K^\top \\ KS^u & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} (S^u)^\top Q (\bar{x} - S^x \bar{x}_1) \\ \mathbf{0} \end{bmatrix}. \quad (19)$$

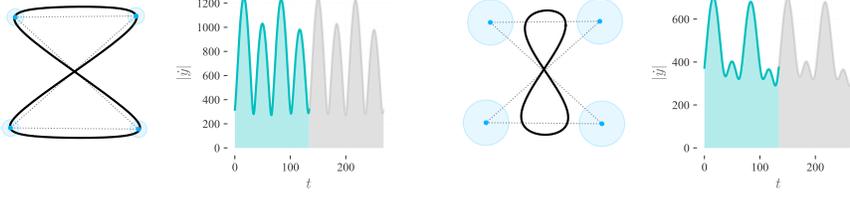


Fig. 5 Periodic motions using Gaussians with different variances. The speed profiles are repeated (in light gray) to visualize the periodicity of the speed profile.

We observe that in order to generate periodic motions that are symmetric, it is convenient to utilize a *wrapped* version of the input components as a new input. To do so we repeat a subsequence of w components at the start and end of the wrapped sequence with the indices of the original sequence organized as follows:

$$[M - w, \dots, M, 1, \dots, M, 1, \dots, w] , \quad (20)$$

where M is the selected number of Gaussians. This produces a new sequence of M° Gaussians for the periodic motion, giving the following passage time sequence:

$$[\tau_{-w}, \tau_{-w+1}, \dots, \tau_{M+1+w}] . \quad (21)$$

This results in a wrapped reference \mathbf{Q} and $\bar{\mathbf{x}}$ that is constructed as described in §3.3. The linear constraint matrix is then given by:

$$\mathbf{K} = \left[\mathbf{0}, \dots, \mathbf{I}, \mathbf{0}, \dots, -\mathbf{I}, \mathbf{0}, \dots \right]^\top . \quad (22)$$

The periodic trajectory is finally computed by plugging the command sequence \mathbf{u} computed with Eqn. 19 into Eqn. 11, and then considering the subset of the trajectory defined between time steps τ_1 and τ_{M+1} .

4 User Interface

The proposed trajectory generation method is efficient and is well suited for interactive design applications. It is easy to drag the centers of the input Gaussians with a typical point-and-click procedure, and it is also easy to interactively manipulate the covariances. For example, this can be done by manipulating an ellipsoid, such that its center defines the mean $\boldsymbol{\mu}_i$, and the axes are used to manipulate the covariance $\boldsymbol{\Sigma}_i$ through its eigendecomposition. The latter can be described with:

$$\boldsymbol{\Sigma}_i = \mathbf{\Theta}_i \mathbf{S}_i^2 \mathbf{\Theta}_i^\top , \quad (23)$$

where Θ_i corresponds to an orthogonal (rotation) matrix, and S_i is a scaling matrix. Here, we describe the 2D case in which the rotation and scaling matrices are given by:

$$\Theta_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}, \quad \theta_i = \tan^{-1} \frac{a_2}{a_1}, \quad S_i = \begin{bmatrix} \frac{\|a\|}{2} & 0 \\ 0 & \frac{\|b\|}{2} \end{bmatrix}, \quad (24)$$

where a and b are the major and minor axes of an ellipse, which can be interactively dragged to manipulate the shape of the distribution (Fig. 6, left). While the examples given are two dimensional, an extension to three-dimensional ellipsoids is straightforward to implement with a so-called *arc-ball* interface [38].

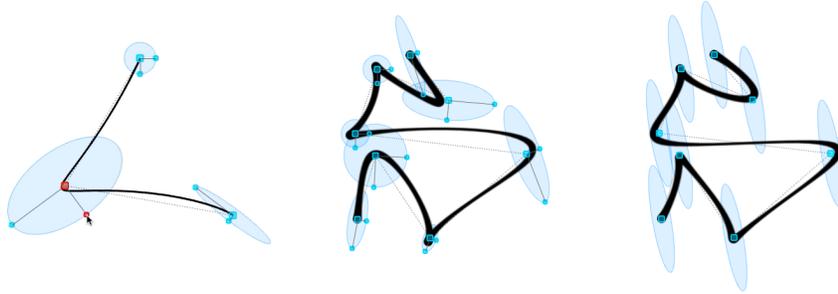


Fig. 6 Examples of user interactions by the manipulation of ellipsoids representing GMMs, and resulting in various animated brush rendering effects.

The trajectories generated by our system are sequences of points, the resolution of which depends on the discretization time step Δt . The distance between consecutive points is not constant and reflects the smooth and physiologically plausible kinematics generated by the model. As a result, it is easy to generate natural looking stroke animations by incrementally sweeping a brush texture along the points of the trajectory [5]. To increase the sense of dynamism, we slightly vary the brush size at a degree inversely proportional to the trajectory speed, which mimics the effect of more ink being deposited on a surface when the movement is slower (Fig. 6).

4.1 Semi-tied structure

In the previous paragraphs, we have seen that it is possible for a user to easily edit the shape and position of each Gaussian. For applications aimed at procedural content generation, it may be desirable to formulate a more parsimonious way of generating trajectories, in which different stylizations are generated without having to specify the covariance of each GMM component. We observe that one convenient way to provide the user this facility is to enforce a *shared orientation* for all covariance ellipsoids, which can easily be achieved with the formulation above by having the

orientations Θ_i set to the same value. This results in a *semi-tied* covariance structure of the input GMMs, in which all covariances share the same eigenvectors but not necessarily the same eigenvalues (Fig. 7).

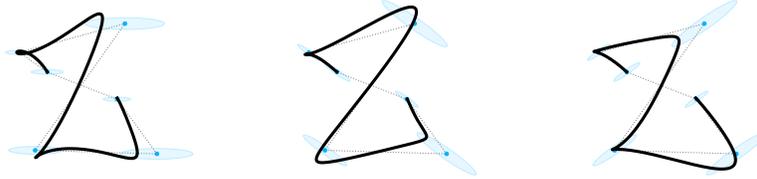


Fig. 7 Three different stylizations of a letter “Z” using semi-tied covariances with different shared orientations.

From a motor control perspective, the semi-tied formalism can be interpreted as the alignment of different movement parts/primitives with a shared coordination pattern [39], which is in line with the hypothesis of postural-synergies at the motor planning level [9]. This implies a shared non-orthogonal (oblique) basis for all the covariances, which produces a shear transformation that in the 2D case transforms a circle into an oriented ellipse. Oblique coordinates have also been suggested to describe the coordination of handwriting movements made with the fingers and wrist only [11], which suggests another possible bio-physical interpretation of this result (Fig. 8).

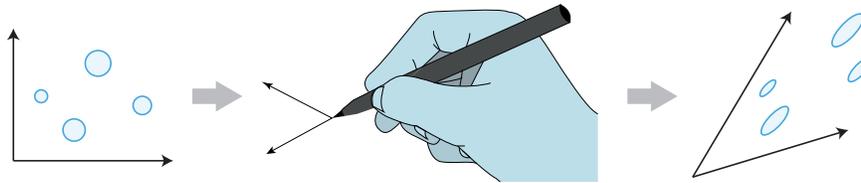


Fig. 8 Illustrative example of how an oblique coordinate system could result from fine movements in handwriting, when using fingers and wrist only.

With this simplified interface, it is possible to explore different stylizations of a key-point sequence with a reduced set of open parameters. The semi-tied covariances enforce a coupling between the coordinates of the trajectory, which results in an observable sense of coordination in the movement. At the same time, minimization of the control command amplitude produces smooth trajectories that evoke a natural drawing movement.

5 Conclusions

We have presented a method for the generation of smooth curves and motion trajectories with a stochastic formulation of optimal control. The output of our method is a trajectory distribution, which describes a family of motion paths that can mimic the appearance and the variability of human-made artistic traces. Each trajectory generated by our method reflects a movement with physiologically plausible kinematics. This can be exploited to produce rendering effects, realistic animations or also to drive the smooth motion of a robotic arm [3]. The input to the method is a sparse sequence of multivariate Gaussians that determine the overall shape of the output and explicitly define its variability. This results in a representation that is similar to the one used in conventional GCAD applications, and that can be edited interactively in a similar manner.

For our use case, we let the user explicitly define the GMM components. However a similar representation can be learned from data with standard maximum-likelihood estimation methods [7]. Our choice of Gaussians as an input and output distribution is principally motivated by its effectiveness and simplicity of representation. From a user-interaction perspective, this allows users to intuitively manipulate the input distributions by modifying the axes of each GMM component ellipsoid (Fig. 6). Furthermore, the straightforward relation of Gaussians to linear systems quadratic error terms allows us to solve the optimal control problem interactively and in closed form, while also offering a stochastic interpretation of the output. Extending the proposed method to non-linear dynamical systems and to distributions other than Gaussians, is an interesting avenue of future research.

The curve generation method presented in this chapter is principally developed with creative computer graphics applications in mind, especially those that require mimicking the visual quality of traces observed in artistic applications of calligraphy and graffiti. There is no specific consensus on a metric that can be used to aesthetically evaluate the quality of visual traces or marks, and for a human this may depend on subjective factors such as cultural and educational background. However, there is growing psychological and neuro-science evidence suggesting that the observation of a static trace resulting from a human-made movement, triggers a mental recovery of the movement underlying its production [35, 20, 29] and that such recovery influences its aesthetic appreciation [28]. As a result, we hypothesize that synthesizing curvilinear traces with kinematics similar to the ones made by a human may trigger similar responses in an observer. Hence, another promising line of future work would be to study the responses of human observers to traces generated with different parameters of the system, as well as evaluate how their artistic expertise or cultural background may influence aesthetic judgment.

Finally, we note that, while in this chapter we focused on the generation of 2D trajectories, the proposed method can naturally be generalized to higher dimensions. We envisage useful future applications, in particular in developing an interface for 3D trajectories, as well as for taking into consideration additional features, such as a drawing tool orientation or the effects of varying the force applied along a trajectory.

References

1. AlMeraj, Z., Wyvill, B., Isenberg, T., Gooch, A., Guy, R.: Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics* **33**(4) (2009)
2. Baran, I., Lehtinen, J., Popović, J.: Sketching clothoid splines using shortest paths. In: *Computer Graphics Forum*, vol. 29, pp. 655–664. Wiley Online Library (2010)
3. Berio, D., Calinon, S., Fol Leymarie, F.: Learning dynamic graffiti strokes with a compliant robot. In: *Proc. of IEEE/RSJ Int'l Conf on Intelligent Robots and Systems (IROS)*, pp. 3981–6. Daejeon, Korea (2016)
4. Berio, D., Calinon, S., Fol Leymarie, F.: Generating calligraphic trajectories with model predictive control. In: *Proc. 43rd Conf. on Graphics Interface*, pp. 132–139. Edmonton, AL, Canada (2017)
5. Berio, D., Leymarie, F.F., Plamondon, R.: Expressive Curve Editing with the Sigma Lognormal Model. In: O. Diamanti, A. Vaxman (eds.) *EG 2018 - Short Papers*. The Eurographics Association (2018)
6. Bernstein, N.A., Latash, M.L., Turvey, M.: *Dexterity and its development*. Taylor & Francis (1996)
7. Calinon, S.: A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics* **9**(1), 1–29 (2016)
8. Cooper, M., Chalfant, H.: *Subway Art*. Holt, Rinehart and Winston (1984)
9. d'Avella, A., Saltiel, P., Bizzi, E.: Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience* **6**(3), 300–308 (2003)
10. De Boor, C.: *A Practical Guide to Splines*, vol. 27. Springer-Verlag New York (1978)
11. Dooijes, E.: Analysis of handwriting movements. *Acta Psychologica* **54**(1), 99–114 (1983)
12. Edelman, S., Flash, T.: A model of handwriting. *Biological cybernetics* **57**(1-2), 25–36 (1987)
13. Egerstedt, M., Martin, C.: *Control Theoretic Splines: Optimal Control, Statistics, and Path Planning*. Princeton University Press, Princeton Oxford (2009)
14. Ferrer, M.A., Diaz, M., Carmona-Duarte, C., Morales, A.: A behavioral handwriting model for static and dynamic signature synthesis. *IEEE transactions on pattern analysis and machine intelligence* **39**(6), 1041–1053 (2017)
15. Flash, T., Handzel, A.: Affine differential geometry analysis of human arm movements. *Biological cybernetics* **96**(6), 577–601 (2007)
16. Flash, T., Henis, E.: Arm trajectory modifications during reaching towards visual targets. *Journal of cognitive Neuroscience* **3**(3), 220–230 (1991)
17. Flash, T., Hochner, B.: Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology* **15**(6), 660–6 (2005)
18. Flash, T., Hogan, N.: The coordination of arm movements. *Journal of Neuroscience* **5**(7), 1688–1703 (1985)
19. Flash, T., Hogan, N.: Optimization principles in motor control. In: *The handbook of brain theory and neural networks*, pp. 682–685. MIT Press (1998)
20. Freedberg, D., Gallese, V.: Motion, emotion and empathy in esthetic experience. *Trends in cognitive sciences* **11**(5), 197–203 (2007)
21. Freeman, F.: Experimental analysis of the writing movement. *Psychological Monographs: General and Applied* **17**(4), 1–57 (1914)
22. Fujioka, H., Kano, H., Nakata, H., Shinoda, H.: Constructing and reconstructing characters, words, and sentences by synthesizing writing motions. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **36**(4), 661–670 (2006)
23. Haerberli, P.: Dynadraw: A dynamic drawing technique. www.graficaobscura.com/dyna/ (1989)
24. House, D., Singh, M.: Line drawing as a dynamic process. In: *Proc. of 15th Pacific Conf. on Comp. Graphics & Applications*, pp. 351–60. IEEE, Maui, USA (2007)
25. Jordan, M., Wolpert, D.: Computational motor control. In: M. Gazzaniga (ed.) *The Cognitive Neurosciences*, 2nd edn. MIT Press (1999)
26. Kyprianiadis, J., Collomosse, J., Wang, T., Isenberg, T.: State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics* **19**(5), 866–885 (2013)

27. Lacquaniti, F., Terzuolo, C., Viviani, P.: The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica* **54**(1), 115–130 (1983)
28. Leder, H., Bär, S., Topolinski, S.: Covert painting simulations influence aesthetic appreciation of artworks. *Psychological Science* **23**(12), 1479–1481 (2012)
29. Longcamp, M., Anton, J.L., Roth, M., Velay, J.L.: Visual Presentation of Single Letters Activates a Premotor area Involved in Writing. *NeuroImage* **19**(4), 1492–1500 (2003)
30. Lu, J., Yu, F., Finkelstein, A., DiVerdi, S.: Helpinghand: example-based stroke stylization. *ACM Transactions on Graphics (TOG)* **31**(4), 46 (2012)
31. McCrae, J., Singh, K.: Sketching piecewise clothoid curves. *Computers and Graphics (Pergamon)* **33**(4), 452–461 (2009)
32. Mediavilla, C., Marshall, A., van Stone, M., Xuriguera, G., Jackson, D.: Calligraphy: from calligraphy to abstract painting. *Scirpus* (1996)
33. Meirovitch, Y., Bennequin, D., Flash, T.: Geometrical invariance and smoothness maximization for task-space movement generation. *IEEE Transactions on Robotics* **32**(4), 837–853 (2016)
34. Morasso, P.: Spatial control of arm movements. *Experimental Brain Research* **42**(2), 223–7 (1981)
35. Pignocchi, A.: How the Intentions of the Draftsman Shape Perception of a Drawing. *Consciousness and Cognition* **19**(4), 887–898 (2010)
36. Plamondon, R.: A Kinematic Theory of Rapid Human Movements. Part I. Biological cybernetics **72**(4), 295–307 (1995)
37. Plamondon, R., O'Reilly, C., Galbally, J., Almaksour, A., Anquetil, É.: Recent developments in the study of rapid human movements with the kinematic theory. *Pattern Recognition Letters* **35**, 225–35 (2014)
38. Shoemaker, K.: Arcball: a user interface for specifying three-dimensional orientation using a mouse. In: *Graphics Interface*, vol. 92, pp. 151–156 (1992)
39. Tanwani, A., Calinon, S.: Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model. *IEEE Robotics and Automation Letters* **1**(1), 235–242 (2016)
40. Thiel, Y., Singh, K., Balakrishnan, R.: Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2d curves. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 383–392. ACM (2011)
41. Todorov, E., Jordan, M.: Optimal feedback control as a theory of motor coordination. *Nature neuroscience* **5**(11), 1226–1235 (2002)
42. Todorov, E., Jordan, M.I.: Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology* **80**(2), 696–714 (1998)
43. Uno, Y., Kawato, M., Suzuki, R.: Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics* **61**(2), 89–101 (1989)
44. Viviani, P., Schneider, R.: A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology: Human Perception and Performance* **17**(1), 198–218 (1991)
45. Zeestraten, M., Calinon, S., Caldwell, D.G.: Variable duration movement encoding with minimal intervention control. In: *Proc. of Int'l Conf. on Robotics and Automation (ICRA)*, pp. 497–503. IEEE, Stockholm, Sweden (2016)