Dynamic Graffiti Stylisation with Stochastic Optimal Control

Daniel Berio
Goldsmiths, University of London
d.berio@gold.ac.uk

Sylvain Calinon
Idiap Research Institute
sylvain.calinon@idiap.ch

Frederic Fol Leymarie
Goldsmiths, University of London
ffl@gold.ac.uk

ABSTRACT

We present a method for the interactive generation of stylised letters, curves and motion paths that are similar to the ones that can be observed in art forms such as graffiti and calligraphy. We define various stylisations of a letter form over a common geometrical structure, which is given by the spatial layout of a sparse sequence of targets. Different stylisations are then generated by optimising the trajectories of a dynamical system that tracks the target sequence. The evolution of the dynamical system is computed with a stochastic formulation of optimal control, in which each target is defined probabilistically as a multivariate Gaussian. The covariance of each Gaussian explicitly defines the variability as well as the curvilinear evolution of trajectory segments. Given this probabilistic formulation, the optimisation procedure results in a *trajectory distribution* rather than a single path. It is then possible to stochastically sample from the distribution an infinite number of dynamically and aesthetically consistent trajectories which mimic the variability that is typically observed in human drawing or writing. We further demonstrate how this system can be used together with a simple user interface in order to explore different stylisations of interactively or procedurally defined letters.

CCS CONCEPTS

•**Computing methodologies** →**Procedural animation;** •**Human-centered computing** →*Gestural input;*

KEYWORDS

Human hand-writing movement modeling; procedural calligraphy; graffiti and tags generation; model predictive control; stochastic optimal control; smoothing splines; iconic and kinemic letter forms.

1 INTRODUCTION

In this study we adopt tools from optimal control, robotics and computational motor control to generate synthetic traces that are visually and dynamically similar to the ones made by a human expert when drawing or writing. We describe a system that enables a user or an algorithm to rapidly define such traces through the specification of a control polygon made of a coarse sequence of targets. The user can then generate and interactively manipulate

Figure 1: Variations on a tag. (a) In (1) is an original tag made with a marker by a graffiti artist; (2) a user rapidly sketches a control polygon by placing points (targets) near curvature extrema in overlay working from an image of the tag; (3) the user adjusts the Gaussians interactively to follow the trace of the original tag; (4) the reproduced tag rendered with a textured brush. (b) Variations on the tag by modifying the parameter d**. (c) Variations on the tag generated from the specified targets using semi-tied covariances (illustrated as orientable yellow ellipses).**

a family of motion trajectories, which follow the target layout and are characterised by dynamics that are similar to the ones that can typically be observed in human hand movements (Fig. 1). The smooth dynamics produced by the system can be exploited to generate natural looking stroke animations, expressive renderings of the trajectory evolution, or drive the smooth end effector motions of a robotic drawing device. In this study we particularly emphasise the applications of our system to the generation of traces that mimic the visual quality of certain forms of calligraphy and graffiti art.

Graffiti, which is also commonly referred to as "writing" or "aerosol art", is an art form that emerged in the late 1960s when it started to appear on the surfaces of the New York City subway [14, 28]. Since then graffiti has developed into a rich and complex art form that revolves around various stylisation and abstractions applied to the letters of an alphabet, and that can be seen today on the walls and surfaces of most urbanscapes around the globe. *Movement* plays two different roles in graffiti. On the one hand, it

is in itself at the source of concern of this contemporary art form, where the surfaces of subway trains serve the role of a *moving canvas*. Graffiti art is also often seen along train tracks and on the walls bordering highways and busy roads, where it is meant to be appreciated while the observer is moving. On the other hand, the mastery of rapid hand gestures is crucial to the aesthetics and style obtained in producing the traces forming the artefacts [19, 36]. In our work we are mainly concerned with the latter aspect involving movement and propose a probabilistic computational framework to model the production of graffiti.

Furthermore, in this study, we focus on the earliest and fundamental type of graffiti art: the highly stylised signature of an artist's pseudonym, commonly referred to as a *tag*. The manner in which a tag is written is commonly referred to as "handstyle" [14] and identifies the artists's personal style and skill. A well executed handstyle is the result of years of practice, and its visual quality is directly related to the spontaneity in which the movements are executed. This is reflected in graffiti jargon with the term *flow*, which denotes the quality of execution of a tag.

We consider the graffiti stylisation of a letter-form with an approach inspired by the work of semiotician *William C. Watt* [47] who studied the evolution of the Latin alphabet with two complementary descriptions of the letter form: one *iconic* where the letter is described in its basic structure as a *sign*, and one *kinemic* — the study of gestures as body language — where the letter is considered as a dynamic representation of the movements that produce its trace on canvas. As an example, Watt demonstrated how the same iconic representation transforms an upper case "A" into a lower case "α" through a process he calls *facilitation*, which is the tendency to reduce effort during the kinemic production of a letter. In our study we explore a similar approach for the synthetic generation of different graffiti handstyles. We define an iconic description of a letter form through a coarse sequence of target loci: the centres of multivariate Gaussian distributions with full covariances. The kinemic realisation of the letter is then produced using a stochastic optimal control formulation, in which a dynamical system is optimised to follow the spatial layout of the targets as well as the coordination patterns defined by the covariances.

With such a probabilistic formulation, the optimisation process results in a ***distribution*** of trajectories [8], rather than a single path. This allows for example to easily capture the subtle variations that can be typically observed in multiple instances of writing or drawing by the same person (§3.6). In addition, varying the shapes of the Gaussians as well as modifying the optimisation and dynamical system parameters, result in different kinemic realisations of the same target sequence (Fig. 1.(c)). This in turn generates different trajectories that are qualitatively similar to different handstyles that can typically be observed in graffiti tags produced by a human artist.

The rest of this paper is organised as follows: after a brief background on related work (§2), we will first provide a detailed description of the optimal control method used to generate trajectories (§3) and then demonstrate how it can be applied for the procedural stylisation (§4) and generation (§4.2) of graffiti tags.

2 BACKGROUND

A rich history of experimental research has brought to light a number of principles that characterise human hand motions, based on dynamic (time, speed) and figural (curvature, shape) aspects. The tangential speed profile of point-to-point aiming movements typically assumes a "bell shape" [17, 33, 38], variably asymmetric depending on the rapidity of the movement [35, 39]. It is generally accepted that complex movements can be described with the superimposition of a discrete number of basic "ballistic" primitives often referred to as *strokes* [34, 41, 43], which are also characterised by bell shaped velocity profiles. With experience, a movement tends to become smoother [39–41] and the number of velocity peaks decreases. This phenomenon is known as *co-articulation* and can be interpreted as the *chunking* of movement primitives at the planning level [41]. The speed of human hand movements tends to be inversely proportional to the trajectory curvature [16, 20]; in certain types of movement, this relation takes the form of a power law [30, 46]. The duration of each movement primitive tends to be similar and independent of the whole movement extent, a phenomenon referred to as local isochrony [27].

Hand movements are typically smooth and appear to obey optimality principles based on the magnitude of high order derivatives of position, leading to various proposed optimisation computational approaches minimising, for example: variance [24], torque [45], "jerk" (or 3rd order derivatives) [18], "snap" (4th order) [15]. In such models, the evolution of a movement is typically defined with point loci along the trajectory that function similarly to spline interpolation points, and which are commonly referred to as *via-points*. A number of models explicitly describe complex motions with the space-time superimposition of ballistic stroke primitives, where the speed profile of a stroke follows a specified bell shaped function, such as a lognormal [37, 38] or a beta function [7]. In such cases, the trajectory evolution is described with a sequence of positions that do not strictly lie along the rendered trajectory, but rather describe the aiming targets of consecutive strokes; these loci are comparable to the control points of a smoothing spline and are commonly called *virtual targets*. Our trajectory description method can be seen as a hybrid between via-points and virtual targets, where Gaussians with low variance behave similarly to via-points, while those with high variance are alike virtual targets.

Egerstedt and Martin [12] use a Hilbert space representation to show that Bézier curves, splines and smoothing splines can be interpreted as solutions to an optimal control problem. In our approach we also solve an optimal control problem by formulating tracking costs as full precision matrices. As such, we may interpret our trajectory generation method as an extension of smoothing splines encapsulating information about precision, coordination and dynamics.

An important number of projects in computer graphics have considered the generation of artistic imagery [29] and the stylisation of line drawings [21, 25, 31]. However, very few such works have exploited motion synthesis techniques; we highlight some of these next. Haeberli [23] created a program that generates calligraphic stylisations of a computer mouse trace on the basis of a mass-spring system. House et al. [26] generate sketchy renderings of a 3D model by using a Proportional Integral Derivative (PID)

controller. AlMeraj et al. [1] mimic the undulation of hand drawn pencil lines by using the minimum jerk model. Berio and Leymarie [6] use the sigma lognormal model [38] to interactively define the motion paths and variations of graffiti tag trajectories.

In a recent companion paper [5], we describe how our method is suitable for interactive applications similar to the popular computer aided design techniques such as Bézier curves and splines; we also provide a more detailed overview of the implementation based on the principles of stochastic optimal control. In the work reported here, we extend the latter method towards generative applications, and focus on the task of trajectory *stylization* inspired in particular by the work of W. C. Watt [47]. We also introduce the use of semi-tied covariances to allow the user to rapidly explore different stylisations in an intuitive manner (Fig. 1.(c)).

3 TRAJECTORY GENERATION

We describe a trajectory by optimising the evolution of a dynamical system controlled by its highest order derivative along the spatial layout of an ordered sequence of multivariate Gaussians. The optimisation is formulated with a cost function that forces the system to track the Gaussians while limiting the amplitude of the control command. The resulting trajectory is smooth up to the order of the dynamical system, and the corresponding dynamics 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. The centres of the Gaussians define a form of control polygon or "motor plan" (in robotics' jargon) that describe the overall spatial evolution of the trajectory. Also, the covariances permit to define the variability as well as directional trends of trajectory segments (Fig. 2).
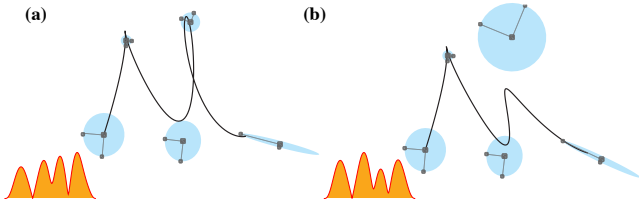

Figure 2: Examples of Gaussian targets (4th order system) with associated speed profiles (bottom left), one (bell shaped) per stroke. Note that in (b) the top variance has been increased, which facilitates the evolution of the trajectory into the slim and elongated covariance at the bottom right.

As previously mentioned, this representation can be seen as a hybrid between the traditionally used via-point and virtual target trajectory descriptors: with a low variance, the trajectory is forced to pass close to the mean of the distribution, which effectively results in a close approximation of a via-point. Using a higher variance produces an effect similar to smoothing splines, with the centers of the Gaussians acting as virtual targets. In addition, non-spherical covariances allow to capture more complex spatial constraints, such as forcing a movement to follow a given direction or to pass through a narrow region of space (Fig. 2). The behaviour of such a system is consistent with the *minimal intervention principle* [44, 48], which

proposes that deviations from an average trajectory are only corrected when they interfere with the required precision of a task. In our case, we locally achieve the required precision by tuning Gaussian covariances.

3.1 Dynamical system

We generate a trajectory with an nth order discrete linear time invariant system defined with the state space form:

$$\xi_{t+1} = A\xi_t + Bu_t, \tag{1}$$

where the state

$$\xi_t = \left[x_t^\top, \dot{x}_t^\top, \dots, \overset{(n-2)}{x_t}{}^\top, \overset{(n-1)}{x_t}{}^\top \right]^\top \tag{2}$$

contains the position and its derivatives up to order $n - 1$, and 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 utilise a chain of n integrators commanded by its highest derivative, with (continuous) system matrices:

$$\bar{A} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix}. \tag{3}$$

The discrete time versions of the system matrices can be computed by using a Zero Order Hold (ZOH) or a simple forward Euler discretisation given by:

$$A = \Delta t \bar{A} + I \quad \text{and} \quad B = \Delta t \bar{B}. \tag{4}$$

3.2 Optimisation

An optimal trajectory of N time steps is computed by minimising a tradeoff between deviations from a desired reference state $\hat{\xi}_t$ (*tracking cost*) and limiting the magnitude of the control commands (*control cost*) with a quadratic cost function:

$$J = \sum_{t=1}^{N} \left(\hat{\xi}_t - \xi_t \right)^\top Q_t \left(\hat{\xi}_t - \xi_t \right) + \sum_{t=1}^{N-1} u_t^\top R_t u_t, \tag{5}$$

where Q_t and R_t are positive semi-definite weight matrices that define the tradeoff between tracking and control penalties for each time step.

This type of optimisation problem is commonly used in process control and robotics applications, where it is known as discrete Linear Quadratic Tracking (dLQT) corresponding to the linear unconstrained case of Model Predictive Control (MPC) [49]. In a typical control setting, these methods would be used to compute an optimal control command for the current time step based on a linearization of the system, and then repeated iteratively for the subsequent time steps. The mathematical framework is however more general and can also be exploited within a planning perspective. In our trajectory synthesis use case, by assuming a system with no disturbance, we can compute all the commands in a single batch optimisation step and generate the resulting trajectory in a rapid manner.

3.3 Tracking cost

We describe a trajectory with an ordered sequence of m multivariate Gaussians $\left\{ \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i) \right\}_{i=1}^{m}$, each defining a state. With an assumption of *local isochrony*, we keep a fixed duration T_s per state, which gives a total trajectory duration of $T = mT_s$ and a corresponding discretised trajectory with $N = T/\Delta_t$ time steps. The tracking weights $\boldsymbol{Q}_t$ and target $\hat{\xi}_t$ for each time step are then formulated by repeating each state T_s/Δ_t times in a stepwise manner along a state vector $\boldsymbol{s} \in \mathbb{N}^N$ (e.g. $\boldsymbol{s} = \{1, 1, 2, 2, 2, 3, \ldots, m\}$) and letting:

$$\hat{\xi}_t = \boldsymbol{\mu}_{s_1} \quad \text{and} \quad \boldsymbol{Q}_t = C^\top \Sigma_{s_t}^{-1} C, \tag{6}$$

with a *sensor* matrix:

$$C = [\boldsymbol{I}, \boldsymbol{0}, \ldots, \boldsymbol{0}], \tag{7}$$

producing zero entries in $\boldsymbol{Q}_t$ for the state derivative terms. This corresponds to a feedback system observing only positions and allowing the specification of states only using position constraints.

This probabilistic formulation of the system states lends itself well for being manipulated interactively or procedurally in an interface that is similar to conventional curve editing techniques such as Bézier curves or splines [5]. A natural interface is then to let the user manipulate an ellipsoid, the axes of which map to the corresponding covariances (Fig. 2). Such covariances can be generated through the eigendecomposition:

$$\Sigma_i = \Theta_i S_i \Theta_i^\top, \tag{8}$$

where Θ_i and $S_i^{\frac{1}{2}}$ correspond in the interface to the rotation and scaling matrices defined by the ellipsoid axes.

3.4 Control cost

The weight matrices $\boldsymbol{R}_t$ define a penalty on the amplitude of control commands. Typically this cost is formulated as a constant diagonal term that is inversely proportionally to the maximum square norm of the control command. In order to achieve approximately equal tracking performance across different system orders (Fig. 3), we express the control cost in terms of a *maximum allowed displacement* d, and compute $\boldsymbol{R}_t$ using the frequency gain of the integrator chain:

$$R_t = \frac{1}{(\omega^n d)^2} \boldsymbol{I} \qquad \text{and} \qquad \omega = 2\pi T_s, \tag{9}$$

where the frequency, ω, is empirically set using the state duration as a period. Lower values of d tend to smooth the trajectory, while higher values generate sharper paths. Because the cost function is defined as a tradeoff between tracking and control cost, it is in practice possible to achieve the same effect by either increasing the variance of the Gaussians or decreasing the value of d.

3.5 Least squares solution

The optimal trajectory can be retrieved iteratively using dynamic programming [5, 9], or in batch form by solving a large regularised least squares problem. Here we describe the latter, which is more compact and allows a straightforward probabilistic interpretation of the result. To compute the least squares solution, we exploit the time invariance of the system, and express all future states as a function of the initial state ξ_1 with:

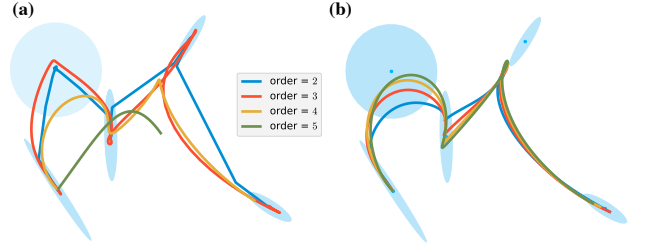$$\xi = S_\xi \xi_1 + S_u \boldsymbol{u}, \tag{10}$$

Figure 3: (a) Trajectories with R_t constant across increasing orders of the dynamical system. (b) Improved tracking consistency by computing R_t depending on the system order and a maximum displacement parameter d.

where

$$S_\xi = \begin{bmatrix} \boldsymbol{I} \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \quad \text{and} \quad S_u = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ B & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ AB & B & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \ldots & B \end{bmatrix}. \tag{11}$$

We then express the cost function (5) in matrix form as:

$$J = (\hat{\xi} - \xi)^\top \boldsymbol{Q} \, (\hat{\xi} - \xi) \; + \; \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u}, \tag{12}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are large diagonal block matrices (with $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ as diagonal block elements), while $\hat{\xi}$, ξ and $\boldsymbol{u}$ are column vectors respectively stacking the reference, state and control commands. Substituting (10) into (12), differentiating with respect to $\boldsymbol{u}$ and setting to zero results in a regularized least squares solution for the command sequence gives:

$$\boldsymbol{u} = \underbrace{(S_u^\top \boldsymbol{Q} S_u + \boldsymbol{R})^{-1}}_{\Sigma_u} S_u^\top \boldsymbol{Q} \left(\hat{\xi} - S_\xi \xi_1 \right), \tag{13}$$

which is then substituted back into (10) to generate a trajectory. From (13) we can see that $\boldsymbol{R}$ effectively acts as a Tikhonov regularisation term (aka ridge regression or weight decay) in the least squares solution, resulting in a smoothing effect on the generated trajectory.

3.6 Stochastic sampling

Because the cost function (12) is a sum of square error terms, its minimisation can be interpreted probabilistically as the product of two Gaussians:

$$\mathcal{N}(\boldsymbol{u}, \Sigma_u) \sim \mathcal{N}\left(S_u^{-1}(\hat{\xi} - S_\xi \xi_1), S_u^\top \boldsymbol{Q} S_u \right) \times \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}), \tag{14}$$

which describes a distribution of control commands with center $\boldsymbol{u}$ and covariance Σ_u. By using the linear relation (10), the distribution in control space can be converted to a *trajectory distribution* (refer to the article by Calinon [8] for details):

$$\mathcal{N}\left(\xi, \Sigma_\xi \right) \quad \text{with} \quad \Sigma_\xi = S_u \Sigma_u S_u^\top. \tag{15}$$

Such a distribution can be used to generate natural variations around the average trajectory ξ (Fig. 4) with:

$$\xi \; \sim \; \boldsymbol{\mu}_\xi + V_\xi \Lambda_\xi^{\frac{1}{2}} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{16}$$

computed from the eigendecomposition:

$$\Sigma_{\xi} = V_{\xi} \Lambda_{\xi} V_{\xi}^{\top}, \tag{17}$$

where V_{ξ} is a matrix containing the eigenvectors and Λ_{ξ} is a diagonal matrix containing the eigenvalues.
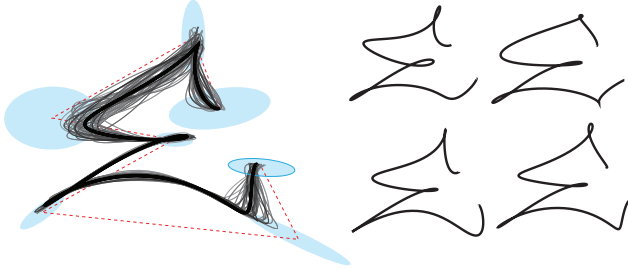

Figure 4: Stochastic sampling of the trajectory distribution. *Left:* **the main trajectory (in black) with random samples from the trajectory distribution (in gray) and the corresponding Gaussians.** *Right:* **randomly selected samples from the same trajectory distribution.**

3.7 Multiple references

The output of the dLQR optimisation procedure can be interpreted as a time varying flow field that depends on the minimisation of the tracking term of the cost function (12). Such a formulation can be extended to additional quadratic costs. This is exploited for example in a robot learning by demonstration application by Calinon [8] to express the cost function in P different coordinate systems. The cost function (12) then becomes:

$$J = \sum_{i=1}^{P} (\hat{\xi}_i - \xi)^{\top} Q_i (\hat{\xi}_i - \xi) + u^{\top} R u \tag{18}$$

and a trajectory of control commands can be retrieved with:

$$u = \left(\sum_{i=1}^{P} S_u^{\top} Q_i S_u + R \right)^{-1} \sum_{i=1}^{P} S_u^{\top} Q_i \left(\hat{\xi}_i - S_{\xi} \xi_1 \right). \tag{19}$$
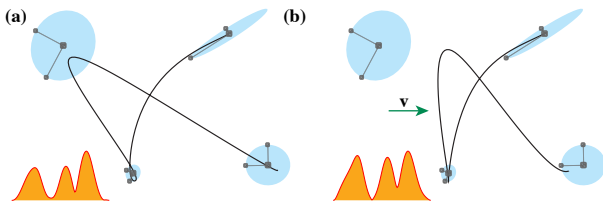

Figure 5: Additional velocity reference (3rd order system). (a) Trajectory tracking four Gaussians. (b) Trajectory generated with an additional velocity reference (v). Note that the optimisation prioritises the Gaussian with low variance, in agreement with the minimum intervention principle [44].

In addition to the position tracking constraints (Fig. 5a), we can add penalties on the velocity terms, such as forcing the trajectory
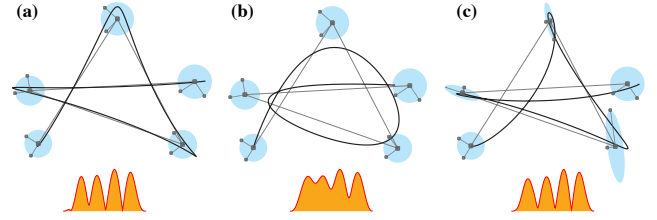

Figure 6: (a) 4th order system following a sequence of 5 targets with spherical covariance. (b) *Facilitation* **[47] using the same system with a lower value of** d. **(c) Effect of non spherical covariance. The respective speed profiles are shown below each trajectory.**

velocity to be under the influence of a constraint vector v by using two references ($P = 2$), where the first tracking constraint is defined similarly as in the previous examples, and where a second tracking constraint is given by

$$\hat{\xi}_{2,t} = 0 \quad \text{and} \quad Q_{2,t} = C_v^{\top} \Sigma_v C_v \quad \forall t \in \{1, \ldots, N\}, \tag{20}$$

with

$$\Sigma_v = v v^{\top} \quad \text{and} \quad C_v = [0, I, 0, \ldots, 0], \tag{21}$$

in which case the sensor matrix C_v corresponds to a feedback system in which only velocity is observed (Fig. 5b).

4 MIMICKING HANDSTYLES

In this section we explore the application of a dual representation of letter form related to the one proposed by Watt [47], in order to mimic computationally the aesthetic and dynamics of graffiti handstyles. In his original formulation, Watt uses a formal grammar to describe the *iconic* representation of a letter in an order/time independent manner. Here, for implementation convenience, we opt for a different approach in which the iconic description of the letter is given by the *centers of a sequence of Gaussians* (or *target sequence*), which consequently also describes a temporal ordering of strokes. The corresponding *kinemic* representation is then given by the covariances associated with each Gaussian as well as the remaining optimisation and dynamical system parameters. The variation of these parameters results in different kinemic realisations of the same target sequence, which we adopt in this study as a *computational* definition of *graffiti handstyle*.

We describe the structure of a letter with sequences of targets, one for each part of the letter made with the pen touching the canvas. Each sequence is made of a low number of targets (usually between 2 and 9), each corresponding to a Gaussian, and can be easily sketched by a user in a point and click procedure. A trajectory generated over a sequence of m targets will typically produce a speed profile characterised by $m - 1$ peaks and (consistently with the inverse speed/curvature relation) local minima corresponding with curvature extrema along the trajectory. To produce a minimal target sequence from an existing letter trajectory, we use an inverse approach and manually (Fig. 1) or automatically (Fig. 10) place targets near salient positions such as curvature extrema [2, 6, 13].

As a simple example of our approach, we mimic the process of facilitation described by Watt. This can be done by specifying spherical covariances for each target and then varying the maximum

displacement parameter d. With a larger value of d the trajectory closely follows the layout of the targets, resulting in a capital "A" (Fig. 6a), while a lower value of d produces a co-articulation effect that produces a trace which is similar to a lower case "a" (Fig. 6b). If we introduce non-isotropic (full) covariances, we can observe that the resulting trajectory follows a more complex and "calligraphic" like evolution (Fig. 6c), which is evocative of effects that can be seen in calligraphy and graffiti produced by humans.

The trajectories generated by our system are sequences of points, the resolution of which depends on the discretisation time step (Δt). The distance between consecutive points is not constant and reflects the smooth and physiologically plausible dynamics generated by the model. As a result, it is trivial to generate natural looking stroke animations by incrementally sweeping a brush texture along the points of the trajectory. 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. 1).

4.1 Semi-tied covariances

In the previous section, we have seen that it is possible for a user to easily edit the shape and position of each Gaussian, where variations of the covariance shape and size result in different kinemic realisations and stylisations of the target sequence. For more generative-oriented applications, it is desirable to formulate a more parsimonious way of generating trajectories that are consistently similar to diverse graffiti handstyles.
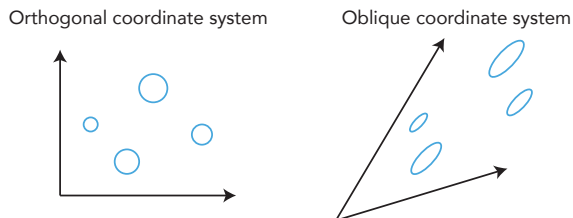

Figure 7: Effect of an oblique coordinate system (right) on the covariances (in turquoise).

In our experiments we observe that one possible way to achieve this result is to enforce a *shared orientation* for all covariance ellipsoids. This is known as *semi-tied* covariances and it corresponds to the case of shared eigenvectors but not necessarily with same eigenvalues. This can be interpreted as the alignment of different movement parts/primitives with a shared coordination pattern [42], which is in line with the hypothesis of postural-synergies at the motor planning level [10]. The tied formalism implies a shared non-orthogonal (oblique) basis for all the covariances. This produces a shear transformation that in the 2D case transforms a circle into an oriented ellipse (Fig. 7). Oblique coordinates have also been suggested to describe the coordination of handwriting movements made with the fingers and wrist [11], which suggests another possible bio-physical interpretation of this result.

In order to manipulate and edit semi-tied covariances, we define a oblique basis H and a magnitude h. The shared covariances can
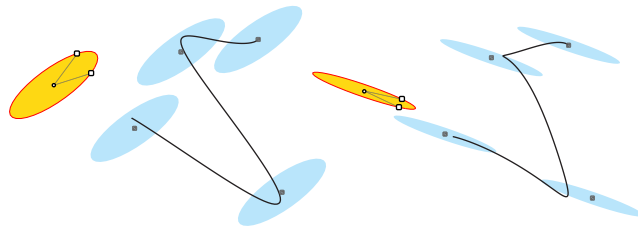

Figure 8: Interface for manipulating semi-tied covariances and corresponding trajectories. The user can drag at the border of the yellow ellipsoid the pair of small black rectangles to redefine the basis vectors of H with magnitude h.

be then computed with:

$$\Sigma_i = HD_iH^\top \quad \text{where} \quad D_i = h_iI \quad \text{and} \quad h_i \sim \mathcal{N}(h, \rho), \quad (22)$$

such that we can perturb h with $\rho > 0$ in order to introduce random variations in the output trajectories. In practice, it is possible to use an arbitrary diagonal matrix for D_i, but we observe that our formulation provides sufficient variety of results and reduces the number of open parameters of the system.

It is then easy to edit the semi-tied covariances with an interface in which the user can drag the basis vectors of H and scale the value of h (Fig. 8). Because the cost function used in the optimisation is given by a tradeoff between tracking and control cost, it is possible to keep the maximum displacement d (which determines the control weight) to a fixed value proportional to the workspace area. The user can then define the smoothness of the generated trajectory by manipulating h, where an increase in h will produce larger covariances and consequently smoother trajectories.

With this interface, a user can interactively explore different stylisations of a target sequence. While the semi-tied covariances enforce a sense of coordination in the movement, the minimisation of the control cost produces smooth trajectories that evoke a natural drawing movement (Fig. 9). A similar method can also be used to generate different stylisation of an input trace, by setting the target loci in correspondence with curvature extrema along the input. We test this approach with the traces of different letters taken from the *UJI Pen Characters Data Set* [32], and observe that the variation of the semi-tied covariance parameters produces trajectories that resemble the original in structure, but possess clearly different handstyles (Fig. 10). We can see in Fig. 10 that our method also allows to easily add smooth ligatures between a character and the next. To do so we simply consider the respective target sequences as a single one, and then remove targets if the angle formed with the previous target and the next is larger than a user-defined threshold.

4.2 Generating Asemic Tags

We have seen how the probabilistic formulation of MPC together with a semi-tied covariance formalism can be used to rapidly explore different stylisations of a letter structure, defined as a coarse sequence of targets. This parsimonious representation can be exploited in combination with procedural generation methods. The user is then left with the simplified task of generating coarse point sequences, while the stylised trajectory evolution is generated by optimal control. Here we demonstrate a simple application, in
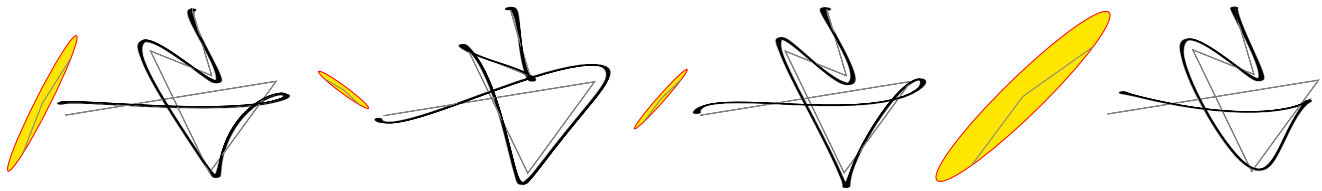
Figure 9: Handstyles for a letter "G" generated with different kinemic realisations (5th order) of the same target sequence (in gray). In yellow, the respective semi-tied covariances.

Figure 10: Handwritten letter stylisation. (a) Traces taken from the UIJI data set and the corresponding target positions (automatically generated). (b) Different stylisations are achieved by using different semi-tied covariance settings. (c) Generation of smooth ligatures. NB: for illustrative purpose we chose a rather low angle threshold of $100°$.

which glyph-like structures (i.e. asemic letters) are generated using a Genetic Algorithm (GA) [22] and then rendered with different styles by optimal control (Fig. 11.(c,d)).

To define a glyph, we produce a set of m random targets, which are generated in polar coordinates by randomly sampling angles and radii values (Fig. 11.(a)). We then use the GA to determine the ordering of the targets by maximising the distance between consecutive loci (Fig. 11.(b), alike an "inverse" Travelling Salesman Problem (TSP)) and rewarding certain stroke directions that might facilitate motor execution by a drawing hand (*e.g.* down and left-to-right strokes). In the presented examples, we use a GA based upon tournament selection and the PMX crossover operator, which preserve ordering and has previously been used to generate approximate solutions for TSP problems [22].
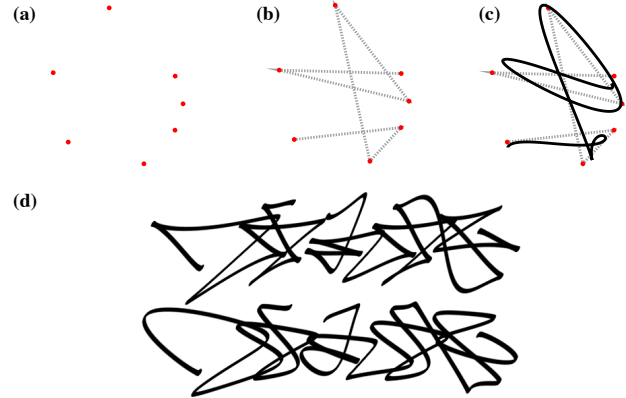

Figure 11: Asemic tags. Top: three steps of generating a random glyph with a GA and MPC using semi-tied covariances. (d) Five random glyphs are concatenated with ligatures to generate patterns evocative of tags with different handstyles.

5 CONCLUSIONS

We have presented a trajectory and curve generation method based on stochastic optimal control that allows the rapid specification of trajectories that are similar to the ones that can be seen in graffiti handstyles. The trajectories generated by our method reflect physiologically plausible dynamics, which can be exploited to generate realistic animations as well as to drive the smooth motion of a robotic arm [4]. We apply our method for the task of trajectory stylisation with a framework inspired by the work of W.C. Watt [47], in which different stylisations are given by the variations of movement that follow a common geometrical structure. In this paper we explore this problem with an optimal control framework, which allows a user to parametrically generate different hand-styles using a simple user interface and few parameters. In a parallel line of work [3], we are exploring the same type of problem with a data-driven approach, in which stylisations are learned using a type of Recurrent Neural Network (RNN).

The framework described in this paper still has a number of limitations and opens the road for a number of future studies. Currently we limit our examples to planar movements, and do not consider parts of the movement that do not touch the surface (e.g. pen up movements). However, the method we describe is directly applicable to higher dimensions. We plan to explore its direct extension to describe 3D movements as well as additional degrees of freedom in drawing movements, such as pen/brush pressure and orientation.

We have demonstrated how using Gaussians with semi-tied covariances introduces an additional element of coordination across the movement, and results in trajectories that are consistent with different instances of graffiti handstyles. However, this method shows limitations when used with letters such as a rounded "O" (Fig. 12.(a)), which currently require a careful interactive manipulation of each covariance (Fig. 12.(b)), or an adjustment of the maximum displacement parameter d.
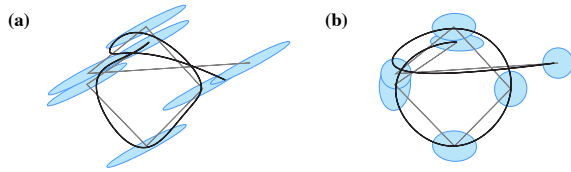

Figure 12: Difficulties in generating a letter "O". (a) Generated with semi-tied covariances. (b) Generated with user-defined covariances.

In the presented examples, we have relied on the qualitative evaluation of a number of expert graffiti artists (including the first author) to set the model hyperparameters. We plan to perform more rigorous aesthetic and cognitive-psychological studies in order to evaluate the visual and dynamic quality of the results and to drive future developments of our methods.

REFERENCES

[1] Z. AlMeraj, B. Wyvill, T. Isenberg, A.A. Gooch, and R. Guy. 2009. Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics* 33, 4 (2009).

[2] F. Attneave. 1954. Some informational aspects of visual perception. *Psychological review* 61, 3 (1954), 183–93.

[3] D Berio, M Akten, F. Fol Leymarie, M. Grierson, and R. Plamondon. 2017. Calligraphic Stylisation Learning with a Physiologically Plausible Model of Movement and Recurrent Neural Networks. In *Proc. of 4th Int'l Conf. on Movement Computing (MOCO)*. London, UK.

[4] D. Berio, S. Calinon, and F. Fol Leymarie. 2016. Learning dynamic graffiti strokes with a compliant robot. In *Proc. of IEEE/RSJ Int'l Conf on Intelligent Robots and Systems (IROS)*. Daejeon, Korea, 3981–6.

[5] D Berio, S. Calinon, and F. Fol Leymarie. 2017. Generating Calligraphic Trajectories with Model Predictive Control. In *Proc. of Graphics Interface (GI)*. Canada.

[6] D. Berio and F. Fol Leymarie. 2015. Computational Models for the Analysis and Synthesis of Graffiti Tag Strokes. In *Workshop on Computational Aesthetics (CAe)*, P. Rosin (Ed.). Eurographics, Istanbul, Turkey, 35–47.

[7] H. Bezine, Adel M. Alimi, and N. Sherkat. 2004. Generation and analysis of handwriting script with the beta-elliptic model. In *Proc. of Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*. IEEE, 515–20.

[8] S. Calinon. 2016. Stochastic learning and control in multiple coordinate systems. In *Intl Workshop on Human-Friendly Robotics*. Genova, Italy, 1–5.

[9] S. Calinon. 2016. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics* 9, 1 (2016), 1–29.

[10] A. d'Avella, P. Saltiel, and E. Bizzi. 2003. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience* 6, 3 (2003), 300–308.

[11] E.H. Dooijes. 1983. Analysis of handwriting movements. *Acta Psychologica* 54, 1 (1983), 99–114.

[12] M. Egerstedt and Clyde Martin. 2009. *Control Theoretic Splines: Optimal Control, Statistics, and Path Planning*. Princeton University Press, Princeton Oxford.

[13] J. Feldman and M. Singh. 2005. Information along contours and object boundaries. *Psychological review* 112, 1 (2005), 243.

[14] A. Ferri. 2016. *Teoria del writing, La ricerca dello stile.* Professional Dreamers.

[15] T. Flash. 1983. *Organizing principles underlying the formation of arm trajectories.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[16] T. Flash and A.A. Handzel. 2007. Affine differential geometry analysis of human arm movements. *Biological cybernetics* 96, 6 (2007), 577–601.

[17] T. Flash and B. Hochner. 2005. Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology* 15, 6 (2005), 660–6.

[18] T. Flash and N. Hogan. 1985. The coordination of arm movements. *Journal of Neuroscience* 5, 7 (1985), 1688–1703.

[19] D. Freedberg and V. Gallese. 2007. Motion, emotion and empathy in esthetic experience. *Trends in cognitive sciences* 11, 5 (2007), 197–203.

[20] F.N. Freeman. 1914. Experimental analysis of the writing movement. *Psychological Monographs: General and Applied* 17, 4 (1914), 1–57.

[21] W.T. Freeman, J.B. Tenenbaum, and E.C. Pasztor. 2003. Learning style translation for the lines of a drawing. *ACM Trans. on Graphics (TOG)* 22, 1 (2003), 33–46.

[22] D.E. Goldberg and R. Lingle, Jr. 1985. Alleles Loci and the TSP. In *1st Intl. Conf. on Genetic Algorithms*. L. Erlbaum Associates Inc., 154–159.

[23] P. Haeberli. 1989. Dynadraw: A dynamic drawing technique. (1989). www.graficaobscura.com/dyna/.

[24] C.M. Harris and D.M. Wolpert. 1998. Signal-dependent noise determines motor planning. *Nature* 394, 6695 (1998), 780–784.

[25] A. Hertzmann, N. Oliver, B. Curless, and S.M. Seitz. 2002. Curve Analogies. In *Proc. 13th Eurographics Workshop on Rendering (EGRW)*. Pisa, Italy, 233–46.

[26] D.H. House and M. Singh. 2007. Line Drawing as a Dynamic Process. In *Proc. of 15th Pacific Conf. on Comp. Graphics & Applications*. IEEE, Maui, USA, 351–60.

[27] M.I. Jordan and D.M. Wolpert. 1999. Computational Motor Control. In *The Cognitive Neurosciences* (2nd ed.), M. Gazzaniga (Ed.). MIT Press.

[28] J. Kimvall. 2014. *The G-word*. Dokument, Stockholm.

[29] J.E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. 2013. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Transactions on Visualization and Computer Graphics* 19, 5 (2013), 866–885.

[30] F. Lacquaniti, C. Terzuolo, and P. Viviani. 1983. The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica* 54, 1 (1983), 115–130.

[31] K. Lang and M. Alexa. 2015. The Markov pen: Online synthesis of free-hand drawing styles. In *Proc. of Workshop on Non-Photorealistic Animation and Rendering (NPAR)*. Eurographics, Istanbul, Turkey, 203–15.

[32] D. Llorens Piñana, F. Prat, Marzal Varo, and others. 2008. The UJIpenchars Database: A Pen-Based Database of Isolated Handwritten Characters. (2008).

[33] P. Morasso. 1981. Spatial control of arm movements. *Experimental Brain Research* 42, 2 (1981), 223–7.

[34] P. Morasso. 1986. Understanding Cursive Script as a Trajectory Formation Paradigm. *Graphonomics* 37 (1986), 137–67.

[35] H. Nagasaki. 1989. Asymmetric velocity and acceleration profiles of human arm movements. *Experimental Brain Research* 74, 2 (1989), 319–26.

[36] A. Pignocchi. 2010. How the Intentions of the Draftsman Shape Perception of a Drawing. *Consciousness and Cognition* 19, 4 (2010), 887–898.

[37] R. Plamondon. 1995. A Kinematic Theory of Rapid Human Movements. Part I. *Biological cybernetics* 72, 4 (1995), 295–307.

[38] R. Plamondon, C. O'Reilly, J. Galbally, A. Almaksour, and É. Anquetil. 2014. Recent developments in the study of rapid human movements with the kinematic theory. *Pattern Recognition Letters* 35 (2014), 225–35.

[39] R. Plamondon, C. O'Reilly, C. Remi, and T. Duval. 2013. The Lognormal Handwriter: Learning, Performing and Declining. *Frontiers in Psychology* 4, 945 (2013).

[40] B. Rohrer and N. Hogan. 2003. Avoiding spurious submovement decompositions: a globally optimal algorithm. *Biological cybernetics* 89, 3 (2003), 190–199.

[41] R. Sosnik, B. Hauptmann, A. Karni, and T. Flash. 2004. When practice leads to co-articulation: the evolution of geometrically defined movement primitives. *Experimental Brain Research* 156, 4 (2004), 422–438.

[42] A.K. Tanwani and S. Calinon. 2016. Learning Robot Manipulation Tasks With Task-Parameterized Semitied Hidden Semi-Markov Model. *IEEE Robotics and Automation Letters* 1, 1 (2016), 235–242.

[43] H.L. Teulings and L. Schomaker. 1993. Invariant properties between stroke features in handwriting. *Acta psychologica* 82, 1 (1993), 69–88.

[44] E. Todorov and M.I. Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature neuroscience* 5, 11 (2002), 1226–1235.

[45] Y. Uno, M. Kawato, and R. Suzuki. 1989. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics* 61, 2 (1989), 89–101.

[46] P. Viviani and R. Schneider. 1991. A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology: Human Perception and Performance* 17, 1 (1991), 198–218.

[47] W.C. Watt. 1988. Canons of alphabetic change. In *The Alphabet and the Brain: The Lateralization of Writing*, D. de Kerckhove and C.J. Lumsden (Eds.). Springer, 122–152.

[48] D.M. Wolpert, J. Diedrichsen, and J.R. Flanagan. 2011. Principles of sensorimotor learning. *Nature Reviews Neuroscience* 12 (2011), 739–51.

[49] M. Zeestraten, S. Calinon, and D. G. Caldwell. 2016. Variable Duration Movement Encoding with Minimal Intervention Control. In *Proc. of Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, Stockholm, Sweden, 497–503.