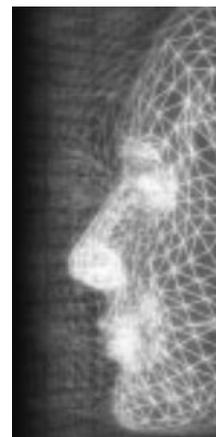


Conditional stochastic simulation for character animation

By N. Courty* and A. Cuzol



In a context of interactive applications, adapting motion capture data to new situations or producing variants of them are known as non trivial tasks. We propose an original method that produces motions that preserve the statistical properties of a reference motion while ensuring some constraints. This method uses principles of conditional stochastic simulation to achieve this goal. Notably, a new real time algorithm, performing sequentially and producing the desired motion is introduced. Possible applications of our method are numerous and several examples are given, along with results. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: character animation; conditional stochastic simulation; motion synthesis

Introduction

The production of new motions from existing ones is a crucial problem in character animation for several reasons: it can alleviate the costs of motion capture and data post-processing; it allows to adapt the motion to distinct types of constraints in a context of interactive applications where there is no *a priori* about the action, and as well can add variety and prevent the clone effect, especially for crowds.¹ Existing solutions for the creation of a new motion can be divided into two categories: interpolative and generative. The first category refers to methods combining (generally in a linear way) existing motions,^{2–4} whereas the second deals with learned models of motions. In the absence of physical or analytical models of motions, statistical models have the capability of expressing the knowledge available in the data, and have revealed over the last years to be a tool of choice for enclosing the motion specific information.^{5–9}

Our method belongs to this category and can synthesize new motions that share the same statistics up to order two of a reference motion. Assuming that the inherent variability of a motion is a realization of a stochastic process, our method first learns its structure by treating it as a Gaussian process. Then, new realizations of motions can be obtained by stochastic simulation, which guarantees

that the obtained motion has the correct statistics. Nevertheless, this is not sufficient to assert the correctness and realism of the motion. The aim of the proposed method is to allow to add kinematic constraints to the system. The contributions of this paper are in this direction and are twofold: (i) using a double kriging operation, we show how it is possible to constrain the stochastic simulation to reach given values at given instants, which amounts to keyframe the simulation (ii) a novel real-time algorithm performing sequentially is proposed to conduct this operation.

The remainder of the paper is organized as follows: the next section presents a short presentation of related work and the following section gives an overview of the method and its philosophy. The subsequent section presents some principles of geostatistics used in our method, notably, the links with Gaussian processes will be emphasized. The section thereafter is dedicated to the presentation of stochastic simulation, along with the algorithmic versant of the theory, followed by possible applications to character animations through three examples: motion reconstruction, variations synthesis, and motion control. The final section concludes the paper.

Background

Our method belongs to the family of statistical models of character motions. The seminal work of Pullen and Bregler¹⁰ is the first to use a non parametric multivariate

*Correspondence to: N. Courty, VALORIA, Université Européenne de Bretagne, Campus de Tohannic, 56000 Vannes Cedex, France. E-mail: courty@univ-ubs.fr

probability density model to express the dependencies between joint angles in motions. Samples drawn from these distributions are then used to generate new sequences from an input motion. Non parametric models have also been used more recently to handle the variation synthesis problem,⁹ where Lau *et al.* use dynamic Bayesian networks to both handle spatial and temporal variations. Our method differs from their work given the fact that in our case only one motion is necessary to produce variants.

However, most of existing works concentrate on parametric families of statistical models. In Reference [11], Brand and Hertzmann were the first to model a motion with hidden Markov models. The motion texture paradigm⁶ uses a two level statistical model, where short sequences of motions (textons) are modeled as linear dynamic system along with a probability distribution of transitions between them. Chai and Hodgins^{12,5} also use linear time invariant models such as autoregressive models to model the dynamic information in the motions. Gaussian processes first served in the computer animation community to perform dimensionality reduction and construct a latent variable model.¹³ Gaussian processes have been also widely used in the context of computer vision.¹⁴ In Reference [7], Wang *et al.* extended the latent space formulation with a model of dynamics in the latent space. Most recent applications of Gaussian processes include motion editing,⁸ motion synthesis of a responsive virtual character¹⁵, and style-content separation.¹⁶ Contrary to these previous works, our method does not require any global optimization procedure as it can perform sequentially, thus making it fully suitable for real time systems, even with a large number of characters such as in a crowd.

Method Overview

Let M be a reference motion. M can be represented as a collection of d dimensional vectors q , each of them parameterizing one configuration of the articulated figure. Usually, those vectors are indexed over time, so that $M = \{q_t\}_{t=1, \dots, T}$. Our method, depicted in Figure 1, starts by applying a dimensionality reduction technique to the data. This part is described in the next subsection, while the basic assumptions and the philosophy of our method are described subsequently.

Data Representation

Before applying our method to motion data, we perform dimensionality reduction on them. The objectives are twofold: (i) working on smaller sets of data while keeping most of the informative part, and (ii) decorrelate the different dimensions of the signal so that it is possible to work on them independently. For this purpose, we choose the Principal Geodesic Analysis (PGA) scheme,¹⁷ which has been recently used in the context of compression of motion data.¹⁸ It can be seen as a generalization of PCA on general Riemannian manifolds. Its goal is to find a set of directions, called geodesic directions or principal geodesics, that best encode the statistical variability of the data. In our case, and conversely to Reference [18], the global translation of the root of the character should be taken into account as an important part of the motion. Similarly to Reference [6], we choose to encode the translation velocity of the root in the vector q which then belongs to the following Lie group $\mathbb{R}^3 \times SO(3)^n$ if n joints parameterize the articular configuration of the character.

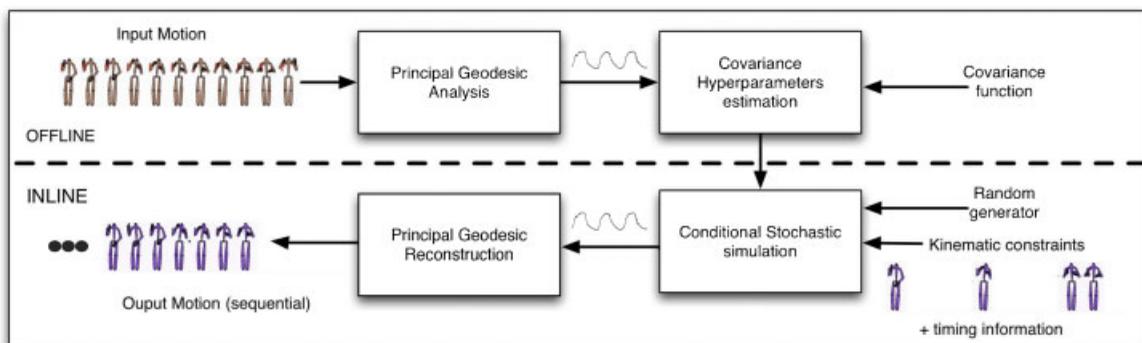


Figure 1. **Overview of the proposed method.** During an offline phase, an example motion is first decomposed with principal geodesic analysis. The resulting trajectories are used to estimate the hyperparameters of a given covariance function. At runtime, the conditional stochastic simulation uses this covariance function, a random generator and some constraints to produce a new motion.

This allows while synthesizing a new motion, to build a new root trajectory by integrating the velocity. The exponential and logarithmic maps for this Lie group are found easily, and as in Reference [18], PGA is computed by applying PCA in the tangent space at the intrinsic mean of the data.

Basic Assumptions

Let $\mathbf{X}_i = \{X_i(t)\}_{t=1,\dots,T}$ be the i th component obtained from the PGA technique applied on the reference motion. This trajectory is assumed to be a realization of a Gaussian process X_i with covariance function C_i . This process is assumed to be ergodic, meaning that its statistical properties can be inferred from one finite realization of it. If the observed realization is of sufficient length, we can indeed consider that it contains the same information as several different realizations of the process. We also assume that the underlying process is stationary, meaning its joint probability distribution does not change when shifted in time, reducing for the Gaussian process to the property that the two first moments do not depend on time.

Motion Synthesis Methodology

Let us first note that the methodology we propose to synthesize new motions requires the explicit knowledge of the covariance function C_i for each resulting component of the PGA decomposition. In practice, a parametric model is first chosen for the covariance function and its hyperparameters are estimated from each realization \mathbf{X}_i . An example of parametric covariance model is the following one:

$$C_i(t, t') = \alpha_i \exp\left(-\frac{|t - t'|^2}{\rho_i}\right) + \sigma_i \delta_{tt'} \quad (1)$$

where ρ_i will be called the length-scale which determines how quickly the covariance falls, δ is one if $t = t'$ and zero elsewhere, and the associated σ_i traduces the nugget effect (small scale variations, corresponding to noise). This model is used for all applications in this paper, and the parameters are estimated with a maximum likelihood approach for each PGA component.

The input of the covariance model and its estimation from observed trajectories in the reduced space correspond to the top right part of Figure 1. Note that this step, like the PGA analysis, is performed offline.

Once the parameters of the covariance functions C_i are known for all PGA directions, a model of motion is available in the PGA space. New motions can then be synthesized from this model. If one aims at simulating motions with the same statistical properties as the reference motion, a realization of a Gaussian process with covariance C_i can easily be obtained for each component, and a new motion can be reconstructed from the PGA approach. However, in order to improve the resulting motion, constraints have to be introduced into the simulation procedure. The problem can then be formulated as the conditional simulation of a Gaussian process, with kinematic constraints as constraint values. The conditional simulation relies on the well known linear prediction problem from Gaussian processes (described in the Section "Prediction from Gaussian Processes"). Based on this linear prediction, it is shown in the Section "Stochastic Simulation" how to respect these constraints while maintaining the statistical properties of the reference motion. This part, performed online, is depicted on the lower part of Figure 1.

Prediction from Gaussian Processes

Given p observations $X(t_1), \dots, X(t_p)$ at times t_1, \dots, t_p of a given Gaussian process with known mean and covariance function C , one can look at the prediction of $X(t)$ for a given time t . In this section, we show that the Kriging approach and the Gaussian Process regression method solve this problem in the same way.

Kriging

Kriging¹⁹ is a linear interpolation method issued from the geostatistical community. Mukai and Kuriyama⁴ used this technique in the context of computer animation to find an optimal set of weights for blending motions. In the kriging approach, the estimation $\hat{X}(t)$ is expressed as a linear combination of the p known values $X(t_1), \dots, X(t_p)$ as follows:

$$\hat{X}(t) = \sum_{i=1}^p \lambda_i(t) X(t_i) \quad (2)$$

where $\lambda(t) = (\lambda_1(t), \dots, \lambda_p(t))^T$ stands for the kriging coefficients.

It is possible to express those coefficients with the following equation:

$$\lambda(t) = \Sigma_{(p)}^{-1} \Sigma_{(t)} \quad (3)$$

where

$$\Sigma_{(p)} = \begin{pmatrix} C(t_1, t_1) & \cdots & C(t_1, t_p) \\ \vdots & \ddots & \vdots \\ C(t_p, t_1) & \cdots & C(t_p, t_p) \end{pmatrix} \quad (4)$$

and

$$\Sigma_{(t)} = (C(t, t_1), \dots, C(t, t_p))^T \quad (5)$$

These coefficients are obtained under the constraints that the estimation is unbiased and that the variance of the kriging error given by

$$\text{Var}(X(t) - \hat{X}(t)) = C(t, t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)} \quad (6)$$

is minimized.

Gaussian Process Regression

A similar approach is known as Gaussian Process (GP) regression in the machine learning community and vision communities.²⁰ The GP approach aims at solving the same prediction problem: given p observations $\mathbf{X}_{(p)} = (X(t_1), \dots, X(t_p))^T$, one looks at the estimation of $X(t)$ at a given unobserved time t . GP's approach solve this problem using the assumption that the process is Gaussian, and building the conditional distribution $p(X(t) | \mathbf{X}_{(p)})$ which is itself Gaussian. The joint distribution of $X(t)$ and $\mathbf{X}_{(p)}$ writes indeed:

$$\begin{bmatrix} \mathbf{X}_{(p)} \\ X(t) \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} \Sigma_{(p)} & \Sigma_{(t)}^T \\ \Sigma_{(t)} & C(t, t) \end{bmatrix} \right) \quad (7)$$

where $\Sigma_{(p)}$ and $\Sigma_{(t)}$ are defined by Equations (4) and (5).

The conditional distribution $p(X(t) | \mathbf{X}_{(p)})$ is then obtained from a little matrix algebra,²⁰ and it comes that this distribution is Gaussian described by

$$p(X(t) | \mathbf{X}_{(p)}) \sim \mathcal{N}(\Sigma_{(t)}^T \Sigma_{(p)}^{-1} \mathbf{X}_{(p)}, C(t, t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)}) \quad (8)$$

The mean $\Sigma_{(t)}^T \Sigma_{(p)}^{-1} \mathbf{X}_{(p)}$ of this distribution is clearly the same as the kriging estimate in Equation (2), and the variance $C(t, t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)}$ corresponds to the variance

of error given by Equation (6). The Gaussian Process regression is then another expression of kriging.

Stochastic Simulation

In the following we assume that Z is a Gaussian process with mean μ and covariance function C . The objective is to simulate trajectories $\mathbf{Z}^{(\text{sim})} = (Z^{(\text{sim})}(t_1), \dots, Z^{(\text{sim})}(t_N))$ of length N of this process. The trajectories have to be independant and respect the statistical properties of Z :

$$E(Z^{(\text{sim})}(t)) = \mu \quad \forall t, \quad (9)$$

$$\text{Cov}(Z^{(\text{sim})}(t), Z^{(\text{sim})}(t')) = C(t, t') \quad \forall t, t' \quad (10)$$

Knowing the covariance function C , the covariance of a trajectory $\mathbf{Z}^{(\text{sim})}$ is then a matrix denoted $\Sigma_{(N)}$ of size $N \times N$, with

$$\Sigma_{(N)} = \begin{pmatrix} C(t_1, t_1) & \cdots & C(t_1, t_N) \\ \vdots & \ddots & \vdots \\ C(t_N, t_1) & \cdots & C(t_N, t_N) \end{pmatrix} \quad (11)$$

Non-conditional Stochastic Simulation

In this section we present how to simulate a trajectory \mathbf{Z}^{NC} respecting the properties (9–10). One possible and simple simulation method is based on the Cholesky decomposition of the covariance matrix $\Sigma_{(N)}$. We first sample a vector $\mathbf{y} = (y_1, \dots, y_N)^T$ composed of N independant realizations of the standard Gaussian distribution, so that $\mathbf{y} \sim \mathcal{N}(0, \mathbf{I}_{(N)})$. Then we set

$$\mathbf{Z}^{\text{NC}} = L_{(N)} \mathbf{y} + \mu \quad (12)$$

where $L_{(N)}$ is obtained from the Cholesky factorization of the covariance matrix: $\Sigma_{(N)} = L_{(N)} L_{(N)}^T$ (provided that $\Sigma_{(N)}$ is positive semi definite). From this decomposition it is easy to verify that $E(\mathbf{Z}^{\text{NC}}) = \mu$ and $\text{Cov}(\mathbf{Z}^{\text{NC}}) = \Sigma_{(N)}$.

One possible concern with this method is, from a computational point of view, the Cholesky factorization of $\Sigma_{(N)}$ which is $o(N^3)$. However, this operation can be conducted only once when $\Sigma_{(N)}$ is known.

Conditional Stochastic Simulation

In some cases, it can be interesting to force the simulations to reach given values $Z(t'_1), \dots, Z(t'_p)$ (experimental data, keyframes specified by animators, etc.) at given time instants t'_1, \dots, t'_p . In this section we explain how to respect these constraints while maintaining properties (9–10).

One could think of simulating new trajectories using the kriging estimate (Equation 2) or sampling from the posterior defined by the GP regression (Equation 8), for all times t between observed values.²⁰ Resulting trajectories would then reach observed values. However, these methods do not create trajectories respecting the property (10). The covariance structure is indeed not respected, and simulated trajectories are then smoother than those simulated with the right covariance structure C .

Note that recently, a method to sample new trajectories solving a global maximum *a posteriori* estimation conditioned to observed values has been proposed by Reference [8]. However, with such an approach there is no guarantee neither that the statistical properties of the reference motion are preserved.

A possible way to obtain trajectories that both respect the required covariance property and reach fixed values is to use a double kriging operation.²¹ Let us recall that the simple kriging allows to find an estimate $\hat{Z}(t)$ at time t that differs from the unknown $Z(t)$ by the kriging error $Z(t) - \hat{Z}(t)$. This error is unknown but can be simulated by means of a secondary process having the same properties as Z . A trajectory $Z^{NC} = (Z^{NC}(t_1), \dots, Z^{NC}(t_N))$ is first simulated using the non-conditional simulation technique described in the previous subsection. A new trajectory $\hat{Z}^{NC} = (\hat{Z}^{NC}(t_1), \dots, \hat{Z}^{NC}(t_N))$ is then obtained by the kriging approach, from all values $Z^{NC}(t'_1), \dots, Z^{NC}(t'_p)$. The resulting kriging error $Z^{NC}(t) - \hat{Z}^{NC}(t)$ for each t is finally added to the trajectory $\hat{Z} = (\hat{Z}(t_1), \dots, \hat{Z}(t_N))$ obtained from the kriging based of the given values $Z(t'_1), \dots, Z(t'_p)$:

$$Z^C(t) = \underbrace{\hat{Z}(t)}_{\text{Kriging estimate}} + \underbrace{Z^{NC}(t) - \hat{Z}^{NC}(t)}_{\text{Kriging error}} \quad \forall t \quad (13)$$

We can directly observe that the trajectory Z^C goes through fixed values $Z(t'_1), \dots, Z(t'_p)$, since the kriged trajectory \hat{Z}^{NC} goes through fixed values $Z^{NC}(t'_1), \dots, Z^{NC}(t'_p)$:

$$Z^C(t'_i) = \hat{Z}(t'_i) + Z^{NC}(t'_i) - \hat{Z}^{NC}(t'_i) \quad (14)$$

$$= Z(t'_i) \quad \forall t'_i = t'_1, \dots, t'_p \quad (15)$$

Moreover, it can be proved that Z^C respects both properties (9) and (10). The resulting simulation is then a sample from a Gaussian process with the required covariance structure C , and that is constrained to go through particular values $Z(t'_1), \dots, Z(t'_p)$.

The algorithm that sums up this conditional simulation technique is the following:

Algorithm 1 Compute trajectory $Z^C = (Z^C(t_1), \dots, Z^C(t_N))$

Input: Covariance structure C of the process

Input: $Z(t'_i)$ at $t'_i = t'_1, \dots, t'_p$

- 1: From C compute the $N \times N$ covariance matrix $\Sigma_{(N)}$
 - 2: $L_{(N)} = \text{Cholesky}(\Sigma_{(N)})$
 - 3: Simulate Z^{NC} using $L_{(N)}$ with Equation (12)
 - 4: Estimate trajectory \hat{Z}^{NC} from $\Sigma_{(N)}$ and fixed values $Z^{NC}(t'_i)$ following the kriging Equation (2)
 - 5: Estimate trajectory \hat{Z} from $\Sigma_{(N)}$ and fixed values $Z(t'_i)$ following the kriging Equation (2)
 - 6: **return** $Z^C(t) = \hat{Z}(t) + Z^{NC}(t) - \hat{Z}^{NC}(t) \quad \forall t = t_1, \dots, t_N$
-

The main computational time is spent in the Cholesky decomposition since this operation is $o(N^3)$. When N is large, this can become a problem. In the context where N is not known, or if a continuous output stream is desired (in order to produce a virtually infinite random sequence), an alternative algorithm can be used. Let us first remark that the Cholesky decomposition produces a matrix L which is lower triangular. This mean that the p th output of the simulation depends on the last $p - 1$ elements that were drawn from the standard Gaussian distribution. This p th output can thus be computed provided that the p th line of L and the past elements are known. However, it is noticeable that the Cholesky decomposition has a recursive formulation, that makes possible to compute the p th line from the $p - 1$ previous lines in the matrix. Also, since the covariance function is assumed to be neglectful after a given distance ρ (corresponding to the length-scale), we can reasonably assume that the influence of known values $Z(t'_i)$ is neglectful whenever $|t'_i - t| < \rho$. By restraining the computation of each element $Z^C(t)$ of the output as a function of sufficiently near $Z(t'_i)$, and by updating iteratively the p th line of the Cholesky decomposition, it is possible to design an algorithm that produces sequentially a correct output:

In this algorithm, `updateCholesky` allows to compute the t th line $L^t_{(\rho)}$ of the Cholesky decomposition from all previous lines.

Algorithm 2 Compute trajectory Z^C sequentially

Input: Covariance structure C of the process

Input: $Z(t'_i)$ at $t'_i = t'_1, \dots, t'_p$

- 1: $y \leftarrow \text{FIFO}(2\rho)$ { y has a FIFO structure of size 2ρ }
 - 2: $Z^{\text{NC}} \leftarrow \text{FIFO}(2\rho)$ {and so Z^{NC} }
 - 3: $t \leftarrow 1$
 - 4: **repeat**
 - 5: $L^t_{(\rho)} = \text{updateCholesky}(L^{0:t-1})$
 - 6: $y \leftarrow \text{push}(y, \sim \mathcal{N}(0, 1))$
 - 7: $Z^{\text{NC}} \leftarrow \text{push}(L^t_{(\rho)}y)$
 - 8: Estimate trajectory \hat{Z}^{NC} from C and fixed values $Z^{\text{NC}}(t'_i)$ (Equation (2)), $\forall t'_i$ such that $|t'_i - t| < \rho$
 - 9: Estimate trajectory \hat{Z} from C and fixed values $Z(t'_i)$ (Equation (2)), $\forall t'_i$ such that $|t'_i - t| < \rho$
 - 10: **return** $Z^C(t) = \hat{Z}(t) + Z^{\text{NC}}(t) - \hat{Z}^{\text{NC}}(t)$
 - 11: $t \leftarrow t + 1$
 - 12: **until** needed
-

Application to Character Animation

We propose here several possibilities to exploit conditional stochastic simulation in the context of character animation. The first example shows how conditional simulation can be used to reconstruct missing or damaged parts of a motion; the second one presents possible applications in motion editing and the last one deals with motion control.

Motion Reconstruction

It is usual with traditional motion capture devices to encounter markers occlusions that alter the quality of the motion reconstruction. With markerless motion capture this problem is even more present as far as the complete pose estimation can fail for a more or less short period of time.²² The objective is to reconstruct the missing parts of the signal. Most of the classical approaches perform linear or spline interpolation between the known parts of the motion. In the case of large holes, those types of interpolation behave badly as they tend to produce a continuous and smooth output which is generally different from the original motion dynamics. Our method first learns the covariance structure on the known parts of the motion and then simulates the unknown part of the motion conditioned to all known single frames.

Figure 2 presents an illustration of the reconstruction for two different hole lengths. One can see that

for small holes, the variability between the different simulations proposed by our method is restrained, and that results are close to a simple kriging interpolation. For longer holes, the variability is bigger and results differ from the kriged solution. Far from observations, the kriging converges indeed toward a mean estimate, flattening the reconstructed part. On the other hand, each of the different trajectories simulated by the conditional approach is statistically coherent with the known part of the motion (which means here that the covariance structure of the whole reconstructed signal is the same than the one learned from the known part). Those proposed solutions might not correspond to the real motion, but can be used as credible, potential solutions.

Motion Variations Synthesis

Our method is able to generate new variants of a motion, and, conversely to Reference [9], with only one example motion. Figure 3 shows four different trajectories of the right hand during a punch motion. Those trajectories correspond to four different simulations obtained from a single punching motion. Figure 4 shows variations obtained from a walking motion. When generating motion variations, one could wish to control the deviation from the original motion. To achieve this, let us first note that the density of constraints on precise zones is directly related to the similarity with the original motion. Another possibility would be to keep unchanged some of the first components of the PGA. In this case, only the remaining components have to be simulated. This can be understood if one considers that the first modes of PGA contain the trend of the motion (as discussed in the Section “Method Overview”), and that the stochastic parts are concentrated on the less meaningful modes.

Exemplary Based Motion Control

We show here how conditional simulation can be efficiently used in the context of motion control. By motion control we mean that, given an exemplar motion, a new motion can be produced along with a set of kinematic constraints, and eventually timing information. Conditional simulation allows to derive an efficient, real-time motion synthesis process by allowing to add kinematic constraints, such as hands or feet positions, to the system, along with timing information. The character pose is solved for by applying PGA-based Inverse Kinematics,¹⁸

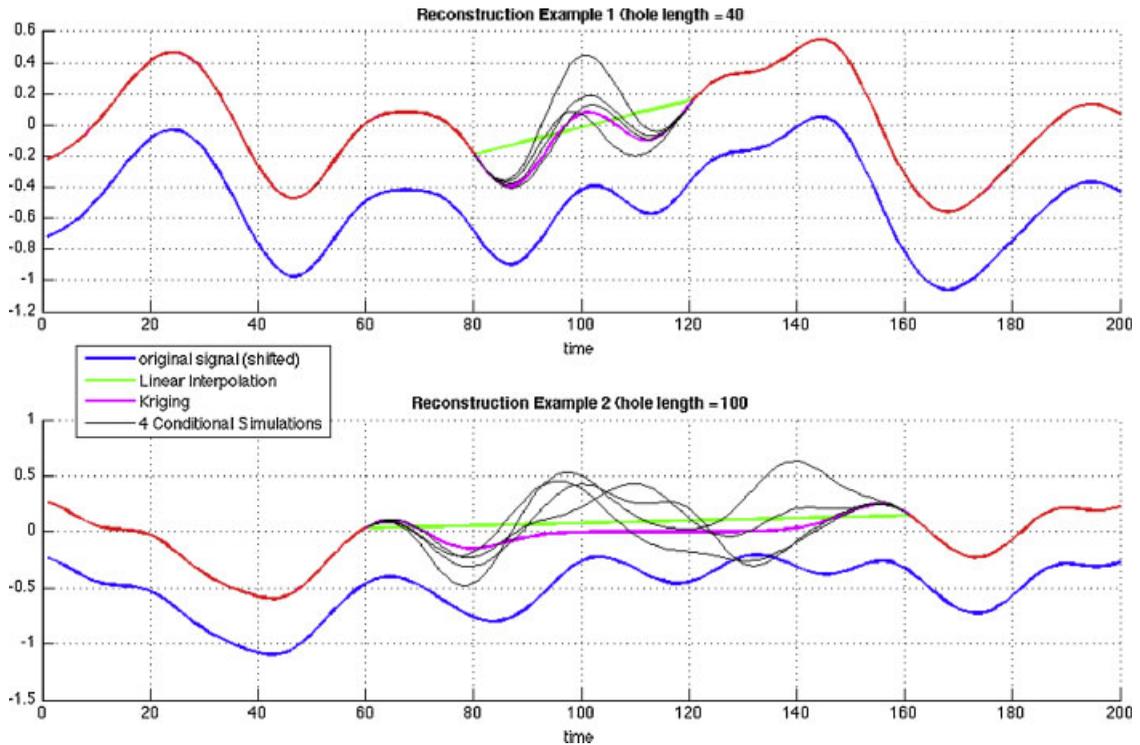


Figure 2. **Hole filling using conditional stochastic simulation.** In this example the length-scale of the covariance function is around 10. When the size of the hole is 40, the simulation is very constrained and the variability is limited. In contrast, when the hole is larger, our method provides different results with a greater variability, whereas the classical linear or kriged interpolate flatten the signal.

which directly gives the corresponding coordinates in the PGA space. Then, a new motion is simulated over a time interval which is centered around the constraint time, and which length is twice the maximum among all estimated length-scales λ_i (which corresponds to the range of time dependence in the covariance model estimated for each PGA component). This interval contains indeed all poses that present significant time dependence with the new constraint and that have then to be recomputed. This simulation is conducted conditioned to every other unchanged poses in the motion. This operation can eventually be processed sequentially.

Figure 5 shows an example of this process. A baseball catch (motion 20 from subject 143 in CMU database) was used. A new catch pose is computed with PGA-based IK (Figure 5a). A new motion is then computed in its vicinity (the first PGA component is shown in Figure 5b). Two image strips showing rendering with a skinned character of both original and simulated sequences are shown (Figure 5c,d). Figure 6 shows another example, where a single kick motion was used to produce a continuous motion of three kicks at three different locations.

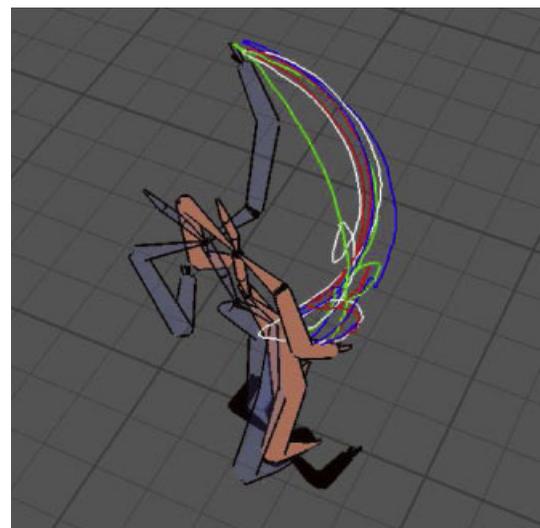


Figure 3. **Generating different punching motions.** The four different trajectories correspond to the right hand position along four different simulations. The transparent punch pose was used as constraint in the simulation process.

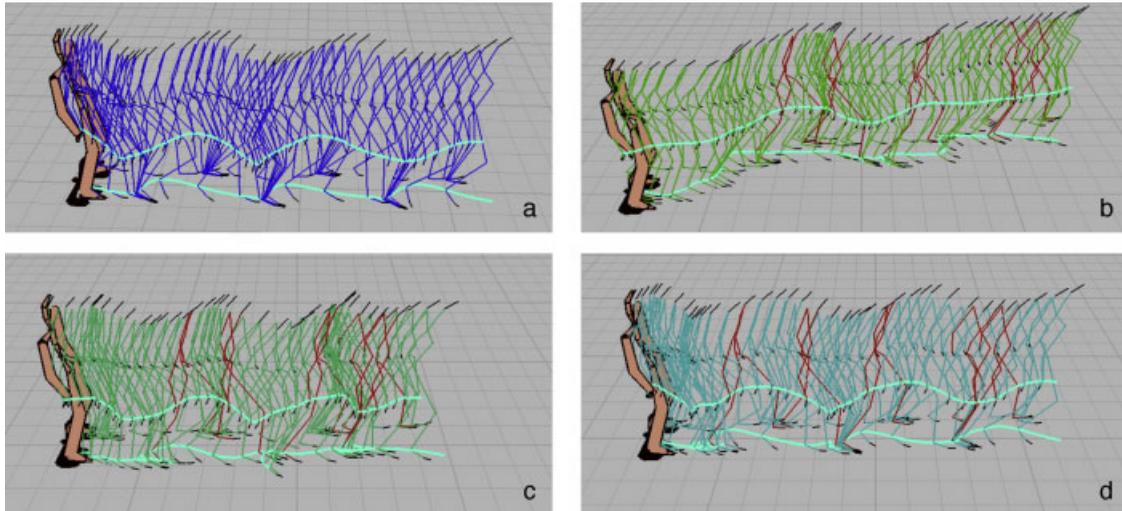


Figure 4. **Variants synthesis.** Our system was used to generate variants of the walking motion presented in (a). Part (b) shows the result of one simulation with pose constraints depicted in a different color (red). Part (c) and (d) are, respectively, obtained keeping unchanged the 3 and 6 first PGA components in the motion.



Figure 5. **Motion control.** This example handles a baseball catch motion. Part (a) presents the original catch and a new catch generated by PGA-IC (applied on both arms). Part (b) shows the first component of the PGA with its new simulated part. Notice the time interval over which the simulation has been performed. Part (c) and (d) illustrate, respectively, the original motion and the synthesized motion on four frames.

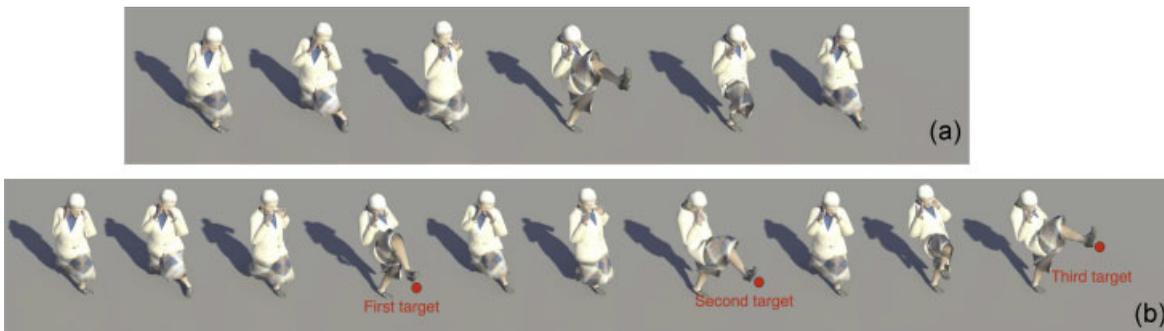


Figure 6. **Kick sequence.** The original sequence (a), containing one kick, is used to produce a continuous sequence (b) of three different kicks at different locations.

Conclusion and Discussion

From one single observed motion, the proposed method based on conditional simulation is able to reconstruct completely new variants of this motion, or to reconstruct unknown parts of it. For all these tasks, the conditional formulation guarantees that the input constraints (defined as known parts of the motion or external kinematic constraints) are respected. Moreover, simulated motion trajectories present by construction of the method the same covariance structure as the reference motion. This property comes from the Gaussian assumption at the root of the method. As a matter of fact, the reference motion is assumed to be a realization of a Gaussian process defined by a mean and a covariance function. The proposed method is able to sample new trajectories that are independent realizations of the same process, and as such have the same statistical properties. Note that this approach is different from a direct sampling based on the posterior distribution described in Equation (8). Such simulations are indeed able to respect kinematic constraints, but do not share the same properties as the reference motion. From a computational point of view, the proposed sequential formulation of the method makes it real-time and as such adapted to interactive applications.

However, it has to be noted that the underlying Gaussian assumption may be too restrictive. The resulting motions may fail to reproduce more complex dynamical structures that could be observed in the reference motion (feet sliding is an example). Moreover, the stationarity assumption is hard to respect for all kinds of motions. Non-stationary components such as clear trends or periodic components for instance, should be removed before learning the covariance model from the data. This can be done for example by fitting a trend and/or periodic model and removing these parts from the reference signal, or trying to adjust non stationary covariance models, but both solutions are far from being trivial. This objective will constitute one of the main follow-ups of this work. A second objective will be the introduction of new types of constraints to the system, related to dynamics information for instance. This will imply to reformulate the conditional simulation part.

ACKNOWLEDGMENTS

Authors thank Professor Shigeru Kuriyama for his insights on this work. This research was partially funded by the pôle MathStic of the University of Bretagne Sud, France.

References

1. McDonnell R, Larkin M, Dobbyn S, Collins S, O'Sullivan C. Clone attack! Perception of crowd variety. *ACM Transactions on Graphics* 2008; **27**(3): 1–8.
2. Rose C, Cohen MF, Bodenheimer B. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 1998; **18**(5): 32–40.
3. Kovar L, Gleicher M, Pighin F. Motion graphs. *ACM Transactions on Graphics* 2002; **21**(3): 473–482.
4. Mukai T, Kuriyama S. Geostatistical motion interpolation. *ACM Transactions on Graphics* 2005; **24**(3): 1062–1070.
5. Chai J, Hodgins JK. Constraint-based motion optimization using a statistical dynamic model. *ACM Transactions on Graphics* 2007; **26**(3): 686–696.
6. Li Y, Wang T, Shum H-Y. Motion texture: a two-level statistical model for character motion synthesis. In *SIGGRAPH 2002, Computer Graphics Proceedings*, 2002; 465–472.
7. Wang JM, Fleet DJ, Hertzmann A. A Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 2008; 283–298.
8. Ikemoto L, Arikan O, Forsyth D. Generalizing motion edits with Gaussian processes. *ACM Transactions on Graphics* 2009; **28**(1): 1–12.
9. Lau M, Bar-Joseph Z, Kuffner J. Modeling spatial and temporal variation in motion data. *ACM Transactions on Graphics (SIGGRAPH Asia)* 2009; **28**(5): 1–10.
10. Pullen K, Bregler C. Animating by multi-level sampling. In *Proceedings of Computer Animation*, 2000; 36–42.
11. Brand M, Hertzmann A. Style machines. In *SIGGRAPH 2000, Computer Graphics Proceedings*, 2000; 183–192.
12. Chai J, Hodgins J. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics* 2005; **24**(3): 686–696.
13. Grochow K, Martin S, Hertzmann A, Popovic Z. Style-based inverse kinematics. *ACM Transactions on Graphics* 2004; **23**(3): 522–531.
14. Urtasun R, Fleet DJ, Fua P. Gaussian process dynamical models for 3D people tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006.
15. Ye Y, Liu CK. Synthesis of responsive motion using a dynamic model. *Computer Graphics Forum* 2010; **29**(2): 1–8.
16. Wang JM, Fleet DJ, Hertzmann A. A multifactor Gaussian process models for style-content separation. In *Proceedings of the International Conference on Machine Learning (ICML)*, June 2007.
17. Fletcher PT, Lu C, Pizer SM, Joshi S. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging* 2004; **23**(8): 995–1005.
18. Tournier M, Wu X, Courty N, Arnaud E, Reveret L. Motion compression using Principal Geodesic Analysis. *Computer Graphics Forum* 2009; **28**(2).
19. Stein ML. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics, New York, NY, 1999.
20. Rasmussen C, Williams C. *Gaussian Processes for Machine Learning*. The MIT Press, Boston, 2006.
21. Lantuéjoul C. *Geostatistical Simulation*. Springer, Berlin, 2002.
22. Moeslund TB, Hilton A, Kruger V. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding* 2006; **104**(2): 90–126.

Authors' biographies:



Nicolas Courty obtained an Engineer degree in Computer Science from INSA Rennes, and a Master degree from University of Rennes I (France) in 1999. He obtained his Ph.D. degree from Insa Rennes in 2002 on the subject of Image-based Animation Techniques. He then spent one year as post-doctoral student in Porto Alegre, Brazil. He integrated the European University of Brittany—UBS (in Vannes, France) as an assistant professor in 2004. His current research interests are analysis/synthesis schemes in computer animation and simulation.



Anne Cuzol was born in 1979. She received the Ph.D. degree in Applied Mathematics from the University of Rennes, France, in 2006. During eight months in 2007, she was a post-doctorate at DIKU, Department of Computer Science, University of Copenhagen. She currently holds a position as assistant professor in the European University of Brittany—UBS, Vannes, France. Her research interests are related to stochastic approaches for motion estimation, tracking, and data assimilation.