EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Improving Sampling-based Motion Control

Libin Liu[1,4] KangKang Yin[2] Baining Guo[3]

[1]Tsinghua University [2]National University of Singapore [3]Microsoft Research Asia [4]University of British Columbia

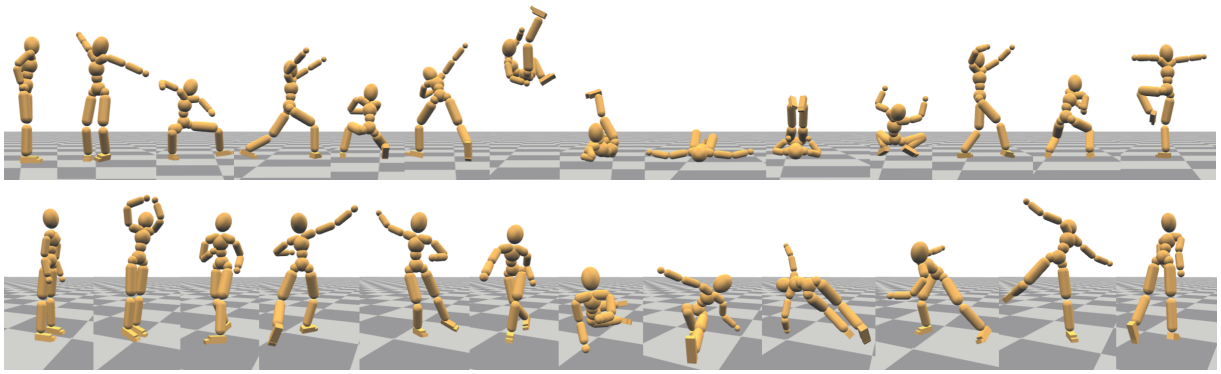

Figure 1: *Open-loop controls reconstructed for a Martial Arts routine: Southern Fist (top); and a Stylized Dance (bottom).*

Abstract

*We address several limitations of the sampling-based motion control method of Liu et at. [LYvdP*10]. The key insight is to learn from the past control reconstruction trials through sample distribution adaptation. Coupled with a sliding window scheme for better performance and an averaging method for noise reduction, the improved algorithm can efficiently construct open-loop controls for long and challenging reference motions in good quality. Our ideas are intuitive and the implementations are simple. We compare the improved algorithm with the original algorithm both qualitatively and quantitatively, and demonstrate the effectiveness of the improved algorithm with a variety of motions ranging from stylized walking and dancing to gymnastic and Martial Arts routines.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The sampling-based motion control method from Liu et al. [LYvdP*10] offers a general framework for constructing open-loop controls from example motions. It is fast, robust, and general enough to handle many types of motions including locomotion and contact-rich floor interactions. Since its introduction four years ago, the method has also been successfully applied to parkour-style highly dynamic motions [LYvdPG12] and skeleton-driven soft body characters [LYWG13].

Despite its successful applications, the original sampling-based control method has several limitations. First, the control reconstruction process treats each trial independently. If failure occurs, the reconstruction restarts without learning anything from the past experiences. Thus when working with challenging motions where the success rate of sampling is low, the method becomes inefficient and requires an excessive amount of reconstruction passes. Second, the original algorithm works on the motion as a whole and always restarts from the very beginning when failure occurs.

Therefore for long motions or motions that contain critical instants, the method rarely succeeds or takes too long to complete. For example, motions with long-flight phases such as gymnastic movements need to be controlled more precisely before the take-off, as very limited control can be done during flight to correct errors. Lastly, sampling-based methods are inherently noisy and thus the resulted motions may look jerky. Such artifacts are not so noticeable for floor interactions but can become disturbing for balancing and locomotion tasks. For instance, the head and body of the virtual character may shake unrealistically during standing or while walking.

In this paper, we address the above limitations of the original algorithm based on a key insight that the later control reconstruction passes should learn from previous successes and failures in order to draw samples in smarter ways. The problems of the original algorithm mainly come from its naive trial-and-error approach to draw samples until a solution is stepped upon by chance. We realize such learning through a modified CMA (Covariance Matrix Adaptation) method as will be described in Section 4. We further integrate a sliding window mechanism into the learning process to improve the reconstruction efficiency and robustness for long motions and motions with critical instants. We also propose two averaging methods to reduce the noise in reconstructed controls, as will be described in Section 5. Both methods are effective in terms of noise reduction, but the simple averaging method needs multiple control trajectories and may become inefficient for challenging motions. In contrast, the elite averaging method averages elite samples during reconstruction and thus requires much less number of trials and time to achieve comparable results.

We compare our algorithm with the original algorithm both quantitatively and visually in Section 6 and in the accompanying video. Lastly, we conclude the paper with discussion on algorithmic limitations and potential areas for future research in Section 7.

2. Related Work

Physics-based motion control has been a topic of extensive research in recent years, ranging from basic balancing [MZS09], locomotion [YLvdP07, MLPP09, LKL10, MdLH10, CBvdP10, KH10], rolling [HYL12, BMYZ13], bicycle stunts [TGLT14], to skilled gymnastic and parkour movements [HWBO95, LYvdPG12, ABdLH13]. However, control construction for highly dynamic stunts and long performance routines remains an open problem. Such motions usually consist of a series of balancing postures, contact-rich moments, highly dynamic segments, and critical instants that are easy to fail. The pioneer work of Hodgins et al. [HWBO95] manually designs controls for athletic maneuvers, which requires human insight and parameter tuning. Ha et al. [HL15] integrate human coaching and numerical optimization to achieve more efficient control design. Feedback policies can greatly enhance control robustness, but so far have mainly been demonstrated for locomotion only [YLvdP07, CBvdP10, LKL10]. Most recently, hand-designed control structures and optimized feedback controls have also been demonstrated for rotational movements such as flips [ABFHdL14]. Generally speaking, designing controls with human guidance requires deep understanding of the motion of interest, which is hard to generalize and often suffers from the lack of styles and details.

Trajectory optimization with respect to spacetime constraints has been investigated in both Computer Animation and Robotics [WK88, SHP04, CH07, WP09, MTP12]. Motion examples are demonstrated to be effective in reducing the dimensionality of the optimization and providing good initial solutions [SHP04, SKL07]. Contact-sensitive behaviors can be discovered through additional contact-related variables [MTP12] or simple goals specified for short spacetime windows [ABdLH13]. When analytical gradients are hard to obtain for the optimization, stochastically estimated gradients can help [KCT*11]. Our method can generate physically valid trajectories just as trajectory optimization methods. The major difference is that we "track" the time-varying control target in the full-body control space through a sampling mechanism. While most trajectory optimization methods directly search for the solution in the trajectory space. Usually such discretization results in search space of higher dimensions and non-differentiable optimization vulnerable to local minima. As a result, no previous trajectory optimization method can produce controls for challenging motions demonstrated in this work at comparable quality.

The sampling-based motion control method of Liu et al. [LYvdP*10] offers a general framework that in theory can work on any type of motions. However, due to the previously mentioned weaknesses of the original algorithm, the success rate of the algorithm is rather low for long and challenging motions, such as a Martial Arts performance or a gymnastic routine. We improve the original algorithm using the technique of distribution adaptation, which is a successful idea applicable to many sampling-based methods, such as Sequential Monte-Carlo [CF00] and Covariance Matrix Adaptation (CMA) [Han06]. Recently CMA has been adopted in a number of animation works in optimizing control strategies [WFH09, WP09, ABdLH13]. Our work utilizes the (μ_W, λ)-CMA-ES method [Han06] in a different way from previous works that optimize the whole control trajectories all at once. We will detail our distribution adaptation method with respect to the conventional way of applying CMA in Section 4.

The idea of learning from past trials has being used in Robotics. For example, the particle filtering technique [DJ11] adapts the distribution of particles according to their rewards. More relevant to our idea is the episodic reinforcement learning method using reward-weighted regressions [HPS09, PK09, KCC10]. Such method iteratively

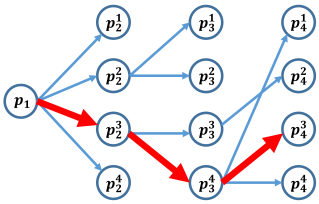

Figure 2: *Schematic illustration of the original sampling-based control reconstruction method. The samples at each iteration form a tree.*

improves the control policy based on past episodes. However, none of these works demonstrate learning in high-dimensional control spaces as ours. Thus adapting such method to highly agile human motions other than basic walking and running remains an open problem.

3. Original Algorithm Review

The original sampling-based control reconstruction method [LYvdP*10] starts with a kinematic reference motion $\tilde{\boldsymbol{m}}$, which is a series of poses in time $\{\tilde{\boldsymbol{p}}_t\}$. We wish to use $\tilde{\boldsymbol{m}}$ to drive a virtual character with PD-servos to perform tasks dynamically. However, due to data noise, model discrepancies between the motion capture subject and the virtual character, and the various simplifications and inaccuracies of PD controls and the simulator, directly tracking $\tilde{\boldsymbol{m}}$ will often fail. For example, the virtual character falls down while walking, or fails to roll over in a rolling task. Thus the algorithm samples necessary corrections $\Delta\boldsymbol{p}_t$ at regular instants so that the compensated target trajectory $\hat{\boldsymbol{m}} = \{\tilde{\boldsymbol{p}}_t + \Delta\boldsymbol{p}_t\}$ not only becomes physically feasible when tracked with PD-servos, but also produces a simulated motion $\boldsymbol{m}$ that is visually similar to the input reference $\tilde{\boldsymbol{m}}$.

We refer to a single run of the control reconstruction method as a *trial* or a *pass*. Each trial is further segmented on the timeline into multiple *iterations*. For iteration k at simulation time $k\Delta t$, we draw N_s samples $\{\Delta\boldsymbol{p}_k^j\}$ from a predefined sampling window. Here Δt is the control construction time step and j is the sample index. Then we advance the simulation from $(k-1)\Delta t$ to the current instant, starting from one of the elite states of the last iteration at time $(k-1)\Delta t$. The n_s offset targets $\tilde{\boldsymbol{p}}_k + \Delta\boldsymbol{p}_k^j$ that result in the best end state are then saved as elite samples of the current iteration. All the elite samples iteratively constitute a K-level tree as shown in Figure 2, where $N_s = 4, n_s = 2$, and $K = 4$. Each node in the tree records the sample $\Delta\boldsymbol{p}_k^j$ and the end state. Finally we assemble $\hat{\boldsymbol{m}}$ by tracing backward in time from the elite samples at the end of the motion and selecting the path with the lowest cost. To further facilitate the reconstruction, $\tilde{\boldsymbol{p}}_k$ can be replaced with a better guess estimated from an inverse dynamics process as suggested in [LYWG13].

The quality of samples are evaluated by a cost function

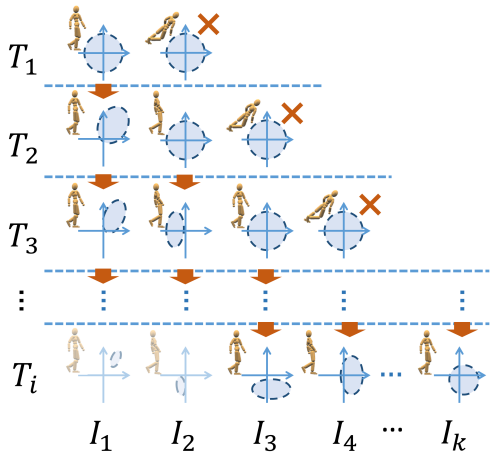

Figure 3: *An illustration of our sample distribution adaptation method. T_1: the first control reconstruction trial fails at iteration I_2, so only the distribution for I_1 is updated. T_2 and T_3: after a few trials, more distributions get updated and longer control trajectories can be constructed. T_i: the distributions of I_1 and I_2 have converged, the reconstruction window is slid to start from I_3.*

that sums up a collection of error terms:

$$E = \frac{1}{c}\sum w_i E_i \tag{1}$$

where E_i's can be tailored to control the full-body poses or end-effectors, and c is a normalization factor. We refer the readers to the original paper [LYvdP*10] for more details.

This algorithm works well for a large variety of motions. However, it has several drawbacks and weaknesses:

- The algorithm always adheres to a simple trial-and-error approach and samples from the start to the end. Thus for long motions or motions containing critical instants such as airborne motions, it either requires an excessive number of reconstruction trials or simply fails.
- The algorithm does not learn from past successes or failures. For each iteration, the samples are always drawn from a zero-centered uniform distribution. In cases where the reference motion needs large compensation, the algorithm has to employ large sampling windows which essentially dilutes the effective sampling density and lowers the success rate of reconstruction.
- In addition to the desirable compensation effect, the samples also introduce undesirable noise which leads to excessive vibration and jerkiness in the simulated motion. This problem is particularly visible in low-dynamic motions such as balancing and walking.

4. Sample Distribution Adaptation

To equip the original algorithm with a learning ability, we employ the idea of distribution adaptation. The key is to ac-

knowledge that even failed reconstruction passes may still contain good samples in the early iterations. We can thus learn from these elite samples to reshape the default uniform distributions from which we draw samples for the next reconstruction trial. Inspired by the successful CMA algorithm, we can move the centers of the distribution as well as update the shape of the distribution to favor samples that are closer to the true solution space.

More specifically, we adapt the sample distributions for earlier successful iterations I_k before we start the next trial T_i, as illustrated in Figure 3. The first trial T_1 draws samples from the default normal distributions $\pi_1^k = \mathcal{N}(\mathbf{0}, \Sigma_1^k)$, where k is the iteration index and Σ_1^k is initialized with the default sampling window size. When T_1 fails at iteration I_2, we update the sample distribution for I_1 using the elite samples and denote the new distribution as $\pi_2^1 = \mathcal{N}(\boldsymbol{\mu}_2^1, \Sigma_2^1)$. Then we start another reconstruction pass T_2 for which we draw samples from the updated distribution π_2^1 for I_1. This process continues and the algorithm utilizes elite samples of past trials to gradually direct future sample drawings closer and closer to the true solution. This is in contrast to the original algorithm which does not learn from past computations and always draws samples randomly and blindly.

We use the (μ_W, λ)-CMA-ES method [Han06] to update sample distributions for each iteration. Our algorithm thus only has one new parameter compared to the original algorithm: the step size parameter for the CMA algorithm, and we initialize it to 0.1 for all the examples shown. The normal distributions are updated according to the quality of elite samples, that is, better elite samples are weighted more during the update. The sample quality is measured by: a) the height of the sample's offspring subtree; and b) the accumulative cost of the best path from the sample to the leaves.

Note that although our distribution adaptation method is inspired by the CMA algorithm, it is different from other applications of CMA in the control optimization literature [ABdLH13]. The previous work samples from the full control space, which usually results in high computational cost, even for sparsely placed control points on the time line. Our problem will become intractable if we were to sample and optimize controls for all the iterations all together. We thus modify the CMA scheme to update distributions for each iteration independently. Such scheme also facilitates easier integration with the original iterative control reconstruction algorithm.

4.1. Sliding Window

The distribution adaptation method described above further enables two improvements to achieve more effective control reconstruction, especially for long motion clips. First, we can choose to run up to a fixed number of iterations in each trial, rather than wait until the trial fails, so that the sample distributions can be updated in a timely fashion. Second, the distributions for the earlier iterations will likely converge after multiple updates, thus we can skip these iterations in future trials to focus sampling only for the later iterations. These two variations combined suggest a sliding window mechanism for control reconstruction.

More specifically, we use fifty iterations (five seconds) as the size of the sliding window in all our experiments. We slide an iteration outside of the window if: (a) the distribution of the iteration has been updated for at least five times and at most twenty times; and (b) the distribution has stabilized, i.e. the cost of the iteration has not decreased for the last five trials; or (c) the best sample drawn from the distribution is good enough according to the cost function. Note that long motions usually consist of both high dynamic segments and static balancing periods, thus we need to normalize the cost function according to the kinetic energy of the window to more accurately estimate convergence.

Algorithm 1 shows the pseudocode of our improved control reconstruction algorithm. Generally speaking, reconstructing controls for a challenging motion with the original algorithm may fail even using big sampling windows, large number of samples, and multiple passes of reconstruction. Yet the improved algorithm can learn from previous trials to advance to success window by window. For motions that can be reconstructed successfully by both algorithms, the reconstructed controls from our method are superior in quality. This is because the improved method is able to gradually move and reshape the sample distributions so that smaller sampling windows can be used to reduce sampling noise.

Algorithm 1 Sample Distribution Adaptation

input: reference motion $\tilde{\boldsymbol{m}} = \{\tilde{\boldsymbol{p}}_k\}$;
default sampling window
output: target trajectory $\hat{\boldsymbol{m}} = \{\tilde{\boldsymbol{p}}_k + \Delta \boldsymbol{p}_k\}$

initialize $\{\pi_1^k\}$
$i = 1$ ▷ trial index
$k_{start} = 1$ ▷ start iteration of the sliding window
repeat
  $k_{stop} = k_{start}$+(size of sliding window)
  for k from k_{start} to k_{stop} **do** ▷ control reconstruction
    draw N_s samples $\{\Delta \boldsymbol{p}_k^j\} \sim \pi_i^k$
    simulate and select n_s samples $\{\Delta \boldsymbol{p}_k^j\}$
    if reconstruction fails **then**
      set $k_{stop} = k$ and **break**
    end if
  end for
  for k from k_{start} to $k_{stop} - 1$ **do** ▷ distribution adaptation
    π_{i+1}^k = update π_i^k with $\{\Delta \boldsymbol{p}_k^j\}$
  end for
  while iteration k_{start} is good enough **do** ▷ window sliding
    $k_{start} = k_{start} + 1$
  end while
  $i = i + 1$
until $\tilde{\boldsymbol{m}}$ is completely reconstructed

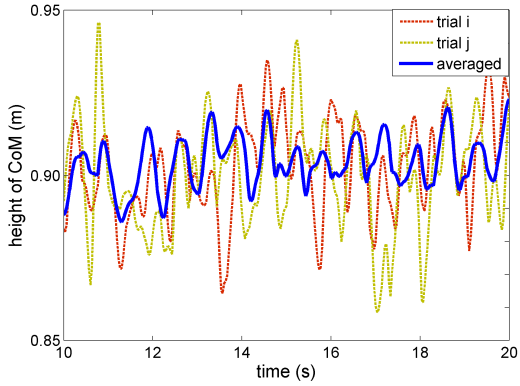

Figure 4: *Noise reduction using simple averaging for stylized walking. The CoM movement is much smoother in the simulation under controls reconstructed from the average of one hundred control trajectories than in the individual simulations (two of which shown as dotted curves).*

5. Noise Reduction

Sampling-based methods are inherently noisy. Post-smoothing of the control trajectory $\hat{\boldsymbol{m}}$, however, does not work as it results in physically invalid controls. We thus follow the law of large numbers, which suggests simple averaging as an effective way for noise reduction. The original sampling-based algorithm can actually utilize averaging straightforwardly as follows. We simply run the reconstruction multiple times until N control trajectories $\hat{\boldsymbol{m}}_i$, $i = 1 \ldots N$ are generated. We denote their corresponding simulated trajectories as $\boldsymbol{m}_i$. Then we average these trajectories as:

$$\bar{\hat{\boldsymbol{m}}} = \frac{1}{N} \sum_{i=1}^{N} \hat{\boldsymbol{m}}_i \tag{2}$$

$$\bar{\boldsymbol{m}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{m}_i \tag{3}$$

Note that these trajectories are a collection of poses in time, so the averaging operation is applied to all the frames at the same instant of time on the trajectories. Since time scaling may occur during control reconstruction as described in [LYvdPG12], the trajectories need to be first resampled to the same length in time before they can be averaged.

We then rerun the sampling algorithm using $\bar{\hat{\boldsymbol{m}}}$ as the initial solution and $\bar{\boldsymbol{m}}$ as the reference. This final reconstruction pass only needs to use very small sampling windows, and thus a much smoother control trajectory can be obtained. Note again that $\bar{\hat{\boldsymbol{m}}}$ itself is not a control solution, just like any kinematically smoothed version of $\hat{\boldsymbol{m}}$, because they violate the equations of motion constraint. However, $\bar{\hat{\boldsymbol{m}}}$ is obtained from averaging physically plausible controls and is thus closer to the solution than kinematically smoothed controls.

Figure 4 shows the effect of noise reduction using the simple averaging strategy on a stylized walking motion. The curves represent the vertical movement of CoM (Center of Mass). The simulation under controls reconstructed from the averaged control $\bar{\hat{\boldsymbol{m}}}$ (shown as blue solid curve) is much smoother than the individual simulations (shown as dotted curves), and thus indicates a stabler walk.

5.1. Elite Samples Averaging

The averaging method described above is simple, effective, and can be directly applied to the original algorithm. However, it requires multiple passes to obtain at least N successful control trajectories $\hat{\boldsymbol{m}}_i$ for the averaging operation. If the success rate of control reconstruction is low, for example for challenging motions that motivated our distribution adaptation algorithm described in the last section, the computation can become prohibitive. We can, however, start to reduce noise by using the elite samples as soon as we have obtained the first control trajectory $\hat{\boldsymbol{m}}_1$.

More specifically, for each sampling iteration of $\hat{\boldsymbol{m}}_1$, i.e., each level of the sampling tree as shown in Figure 2, we select and average the elite samples according to their goodness. We compute a weighted average of all the elite samples $\Delta \boldsymbol{p}_k^j$ at iteration k as follows:

$$\overline{\Delta \boldsymbol{p}_k} = \frac{\sum h_j \Delta \boldsymbol{p}_k^j}{\sum h_j} \tag{4}$$

where h_j is the sample's subtree height and has to be larger than four for a sample to be considered elite. It measures the quality of samples by their descendants, and is therefore better than the near-sighted sample cost function. We then use the averaged samples as an improved initial guess for another control reconstruction pass.

We also shrink the sampling windows gradually and replace the reference motion with the simulated motion periodically. This is because the average of elite samples is a better initial guess for control reconstruction, and the simulated motion is a dynamically-filtered version of the reference motion. This is similar to applying Equations 2 and 3 in the simple averaging scheme. For all our experiments, we shrink the sampling window by 30% after each successful trial. The algorithm is not sensitive to this parameter to be successful, and will just take longer if a smaller shrinking factor is used. We replace the reference motion with the simulated motion of the last successful reconstruction whenever we encounter a failure in the reconstruction process. The above described scheme learns from past successful trials as fast as possible rather than wait for multiple successful reconstructions. Therefore, the sampling noise can be reduced much faster as compared to the simple averaging scheme.

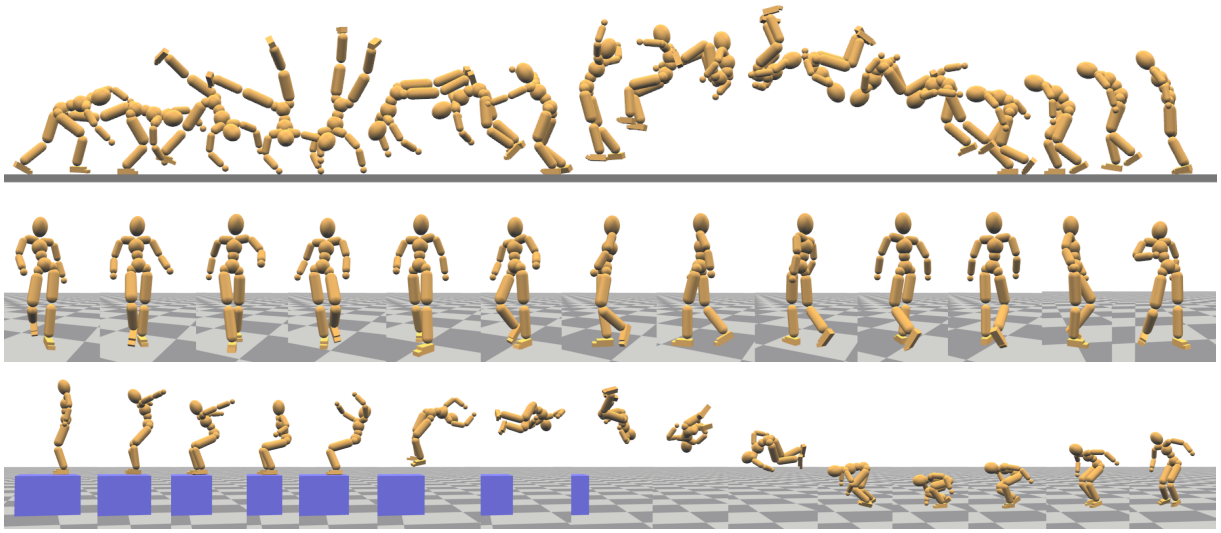

Figure 5: *Simulations of a gymnastic flip (top), a stylized walk (middle), and a twist flip (bottom).*

6. Results

Performance We have tested the improved sampling-based motion control algorithm using a variety of motions, including low-dynamic motions, high-dynamic motions, airborne motions, and long performance routines. Table 1 shows its performance measured on a small 80-core cluster. We initialize the sampling window ω_0 to 0.1, which is then gradually adapted after each trial. To quantitatively measure the noise level, we define NSR (Noise-to-Signal Ratio) for the simulated motion as

$$\mathrm{NSR} = \frac{\sigma^2(\boldsymbol{m} - \tilde{\boldsymbol{m}})}{\sigma^2(\tilde{\boldsymbol{m}})} \times 100 \qquad (5)$$

where $\boldsymbol{m}$ is the simulation trajectory, and $\tilde{\boldsymbol{m}}$ is the reference motion. We compute the motion variance σ using joint rotations. This NSR reflects the fact that visually perceivable noise level correlates with the energy of motions. That is, highly dynamic motions can in a way conceal noise to a certain degree.

We also encourage the readers to evaluate the motion quality through the accompanying video demos. Here in the paper Figure 5 shows the animation strips of a gymnastic flip on the top, a stylized walk in the middle, and a twist flip at the bottom. Figure 1 shows a Chinese Martial Arts routine called Southern Fist on the top, and a stylized dance at the bottom. Our method can also physically retarget motions to different characters with ease, thanks to its robustness. Figure 6 shows the stylized walk retargeted to a character with its leg ratio significantly modified, and a monster character quite different from humans.

Comparison of Reconstruction Table 2 shows a comparison of the sample distribution adaptation algorithm with the original algorithm. Where possible, we assign parameters with same values as in Liu et al. [LYvdP*10] for ease of comparison, such as for the sampling windows and the PD-servo gains. As many of our tested motions were not examined by the original algorithm, at times we do need to augment the original cost function in Equation 1 with new terms for better control reconstruction. More specifically, for all the flipping motions, we augment the cost function with error terms to control the overall body orientation and end-effector positions. Also for all motions with long durations of quiescent standing, we modify the balance term in the original cost function to track the center of the support polygon in order to improve the stability of static balancing.

motion	duration (s)	# trials	time (min)	NSR
Stylized Walk	24.0	45	36.2	2.94
Long Fist	7.6	26	9.1	5.23
Backward Flip	3.0	16	3.2	18.9
Southern Fist	54.0	188	180	6.02
Gymnastic Flip	2.2	27	8.1	22.5
Twist Flip	3.5	8	2.3	27.0
Dribble	22.0	150	93.3	6.77
Dance	64.0	223	198	4.06

Table 1: *Performance of the distribution adaptation algorithm. Timing is measured on an 80-core cluster. The # trials column shows the total number of control reconstruction trials. NSR represents the noise-to-signal ratio of the simulated motion.*

Table 2 shows that the original method usually cannot reconstruct controls successfully when small sampling win-

motion	method	N_s	ω_0	# trials	time (min)	NSR
Long Fist	original	20000	0.5	1	9.3	11.3
		2000	0.5	6	4.0	12.8
		2000	0.1	failure		
	ours	2000	0.5	9	5.4	12.7
		2000	0.1	26	9.1	5.23
Gym Flip	original	20000	0.5	failure		
		2000	0.5			
		2000	0.1			
	ours	2000	0.5	12	3.5	36.3
		2000	0.1	27	8.1	22.5

Table 2: *Comparison of our distribution adaptation algorithm with the original algorithm. Our improved algorithm may take more trials due to the sliding windows, but the total time used is on par with the original algorithm when both worked. Note that tuning parameters of the original algorithm cannot address all failures or improve the motion quality to a satisfactory level.*

motion	method	time (min)	NSR_0	NSR
Stylized Walk	simple	118	12.8	2.75
	elite	17.6		2.68
Long Fist	simple	53.9	7.21	5.02
	elite	5.9		5.78
Backward Flip	elite	2.1	27.8	15.2
Southern Fist	elite	34.4	6.02	4.30
Dance	elite	36.4	4.06	3.70

Table 3: *Noise reduction using the simple and elite averaging schemes. NSR_0 and NSR measure the noise-to-signal ratio of a simulated motion before and after noise reduction respectively. Note that for the latter three motions we did not experiment with simple averaging because the computational cost is prohibitive.*

dows were used. Enlarging the sampling windows ω_0 and increasing the number of samples N_s can help in some cases, but not for all motions such as the gymnastic flip. The motion quality is also low as suggested by the high NSR values. In contrast, our improved method can generate high-quality controls and motions using the same set of parameters ($\omega_0 = 0.1, N_s = 2000$) for all motions. Our algorithm may take more trials due to the sliding windows, but the total time used is on par with the original algorithm when both worked.

Comparison of Noise Reduction Table 3 shows the effectiveness of the simple and elite averaging methods for noise reduction. For simple averaging, we use twenty successful reconstructed control trajectories. For elite averaging, we run a variable amount of trials until we have achieved satisfactory noise reduction. The table indicates that elite averaging usually takes much less total time to achieve comparable results as simple averaging. In addition, for challenging motions such as Backward Flip, Southern Fist, and Dance, we only applied elite averaging as simple averaging becomes computationally too expensive.

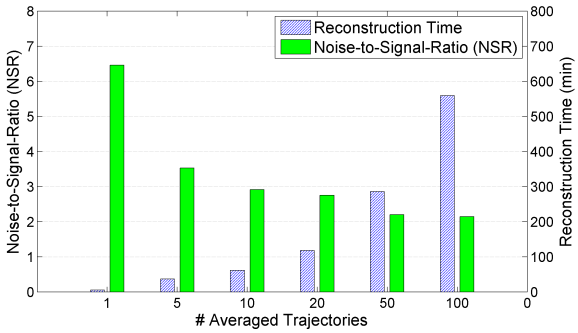

Figure 7: *Simple averaging for Stylized Walk using different numbers of control trajectories. The green bars show the noise-to-signal ratio of the final simulated motion, and the blue stripes indicate the total reconstruction time.*

Figure 7 shows the relationship of NSR with respect to the number of successful control trajectories used in simple averaging. As we can see, although averaging more trajectories can gradually improve the motion quality, there is an upper bound beyond which no noticeable noise reduction can be gained further more. The sweet spot is around twenty trajectories where good noise reduction can be achieved in affordable time. We also encourage the readers to watch the accompanying video to visually judge the effect of noise reduction.

Summary The improved control reconstruction algorithm using the sample distribution adaptation method coupled with the sliding window scheme is more successful for challenging motions, requires less parameter tuning, and produces motions of better quality, compared with the original algorithm. Its performance depends on the number of samples N_s drawn for each iteration, the length of the motion, and the number of reconstruction passes needed. For example, we can use ($N_s = 2000$) to successfully reconstruct the 54-second Southern Fist motion with 188 trials in 3 hours on the cluster. The simulated motions are generally good enough visually, but may still contain visible noise for motions with long static balancing periods where the absolute value of visually tolerable noise level is low. In such cases, we can apply the elite averaging method to further reduce the noise level in just a few additional control reconstruction runs.

7. Discussion

In conclusion, we have successfully improved the sampling-based motion control algorithm of Liu et al. [LYvdP*10], in terms of both robustness and motion quality. More specifically, we can now generate open-loop controls for motions that are computationally intractable to handle with the orig-

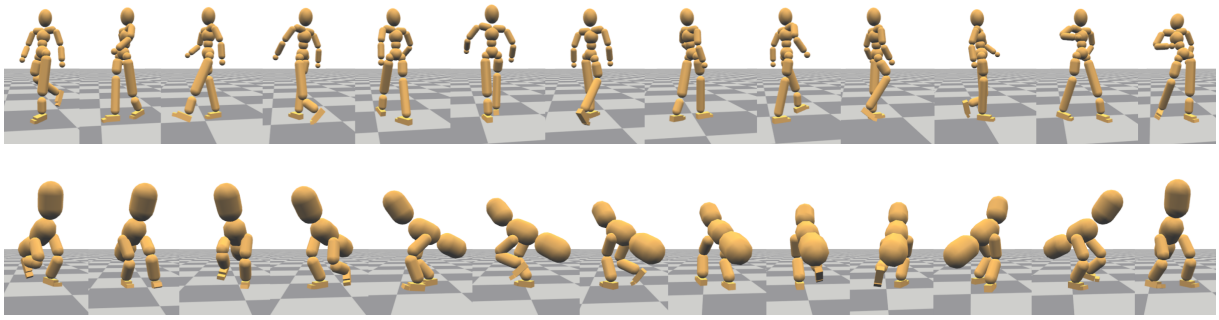

Figure 6: *Stylized walk retargeted onto a character with modified leg ratio (top) and a monster character (bottom).*

inal algorithm; and significantly reduce the noise of the reconstructions. In addition, our ideas are intuitive and the implementations are simple. The original algorithm can still be used for control reconstruction for short and easy motion clips, afterwards simple averaging can be applied to reduce noise. However, for long and challenging motions, all the components presented in this paper are needed. That is, the distribution adaptation approach coupled with the sliding window scheme for control reconstruction, and elite averaging for noise reduction. Control reconstruction employing only the distribution adaptation procedure without sliding windows can easily get stuck in local minima; and using sliding windows alone without adapting the sample distributions is more or less still a trial-and-error approach similar to the original algorithm, and thus suffers from the same problems of low success rate and large noise for challenging motions.

Since it is intractable to sample densely in the high dimensional control space, random noise and reconstruction failures are inevitable in sampling-based control methods. Our strategy is to approach the solution space as close as possible, so that both the success rate and the reconstruction quality can be improved with a good initial guess and small sampling windows. Our key insight is to utilize past computations to direct future sampling. Specifically, the distribution adaptation method progressively improves the solution estimation by updating the sample distributions based on past successes and failures; and the noise reduction methods directly improve the solution from past successes. These strategies are shown to be effective for a variety of motions investigated in this paper.

The distribution adaptation method can be tweaked to reshape the sample distributions more aggressively to further reduce noise, at the cost of more computations. This may result in over-fitting at times, which will then cause the reconstruction being trapped in local minima. Thus we use a relatively low limit for the maximal number of trials that can be performed for the sliding windows, and apply an optional elite averaging postprocessing pass to achieve better quality if needed. In a way, the elite averaging method can be viewed as a manually guided process of adapting the sample distributions. The averaging-based methods, however, are just linear operations in highly nonlinear control spaces and thus cannot completely remove all noise, especially when the controls were constructed with large sampling windows. In such cases, the distribution adaptation method is necessary to guide the initial solution to the true solution space in order for the control reconstruction and averaging methods to be more effective.

Our tested motions were either downloaded from public motion capture databases or captured by our own motion capture system. Thus the motion capture subjects were all different and we simply replayed them on our human model as the reference. Therefore our example motions contain model mismatching errors in addition to the usual motion capture noise. Our reconstruction algorithm is robust enough to automatically perform the necessary "physics-based retargeting". Our additional retargeting experiments also show that the algorithm is robust to more aggressive retargeting tasks. The tracking nature of the reconstruction algorithm, however, implies that the algorithm can fail when the physical accuracy of the example motion becomes too poor, especially during long airborne phases of the motion. One possible solution is to preprocess the reference motion to first enhance its physical realism, e.g., dynamics filtering [YN03, TK05] or physics-based retiming [MPS06]. Some of the algorithm parameters can also be computed using more principled methods, such as using [NF02, LHP05] for setting the joint PD-servo parameters.

For future work, we would like to experiment with building closed-loop controls on top of the output of the improved algorithm, similar to Ding et al. [DLvdPY12] that builds feedback policies for the original algorithm. We are interested in better measures of noise-to-signal ratio for human motions, as our NSR does not always align with human visual perceptions. We also wish to investigate noise reduction in the more general framework of stochastic optimization for motor control.

Acknowledgements: This work was partially supported by Singapore Ministry of Education Academic Research Fund

Tier 2 (MOE2011-T2-2-152). We would like to thank the anonymous reviewers for their constructive comments. We wish to thank Michiel van de Panne for his support for this project. Lastly, we are grateful to Chow Chin Ming's timely help whenever our machines malfunctioned.

References

[ABdLH13] AL BORNO M., DE LASA M., HERTZMANN A.: Trajectory optimization for full-body movements with complex contacts. *TVCG 19*, 8 (2013), 1405–1414. 2, 4

[ABFHdL14] AL BORNO M., FIUME E., HERTZMANN A., DE LASA M.: Feedback control for rotational movements in feature space. *Computer Graphics Forum 33*, 2 (2014). 2

[BMYZ13] BROWN D. F., MACCHIETTO A., YIN K., ZORDAN V.: Control of rotational dynamics for ground behaviors. In *SCA* (2013), ACM, pp. 55–61. 2

[CBvdP10] COROS S., BEAUDOIN P., VAN DE PANNE M.: Generalized biped walking control. *ACM Trans. Graph. 29*, 4 (July 2010), 130:1–130:9. 2

[CF00] CHENNEY S., FORSYTH D. A.: Sampling plausible solutions to multi-body constraint problems. In *Proceedings of SIGGRAPH* (2000), pp. 219–228. 2

[CH07] CHAI J., HODGINS J. K.: Constraint-based motion optimization using a statistical dynamic model. In *Proceedings of SIGGRAPH* (2007), ACM. 2

[DJ11] DOUCET A., JOHANSEN A. M.: A tutorial on particle filtering and smoothing: Fifteen years later. In *Handbook of Nonlinear Filtering*. Oxford, UK: Oxford University Press, 2011. 2

[DLvdPY12] DING K., LIU L., VAN DE PANNE M., YIN K.: Learning reduced-order feedback policies for motion skills. *Tech. Rep. TR-2012-06, University of British Columbia* (2012). 8

[Han06] HANSEN N.: The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, vol. 192 of *Studies in Fuzziness and Soft Computing*. Springer Berlin Heidelberg, 2006, pp. 75–102. 2, 4

[HL15] HA S., LIU C. K.: Iterative training of dynamic skills inspired by human coaching techniques. *ACM Trans. Graph.* (2015). 2

[HPS09] HACHIYA H., PETERS J., SUGIYAMA M.: Efficient sample reuse in EM-based policy search. In *Machine Learning and Knowledge Discovery in Databases* (2009), vol. 5781 of *Lecture Notes in Computer Science*. 2

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of SIGGRAPH* (1995), ACM, pp. 71–78. 2

[HYL12] HA S., YE Y., LIU C. K.: Falling and landing motion control for character animation. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 155:1–155:9. 2

[KCC10] KORMUSHEV P., CALINON S., CALDWELL D.: Robot motor skill coordination with EM-based reinforcement learning. In *IROS* (2010), pp. 3232–3237. 2

[KCT*11] KALAKRISHNAN M., CHITTA S., THEODOROU E., PASTOR P., SCHAAL S.: STOMP: Stochastic trajectory optimization for motion planning. In *ICRA* (2011), pp. 4569–4574. 2

[KH10] KWON T., HODGINS J.: Control systems for human running using an inverted pendulum model and a reference motion capture sequence. In *SCA* (2010). 2

[LHP05] LIU C. K., HERTZMANN A., POPOVIC Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph. 24*, 3 (2005), 1071–1081. 8

[LKL10] LEE Y., KIM S., LEE J.: Data-driven biped control. *ACM Trans. Graph. 29*, 4 (July 2010), 129:1–129:8. 2

[LYvdP*10] LIU L., YIN K., VAN DE PANNE M., SHAO T., XU W.: Sampling-based contact-rich motion control. *ACM Trans. Graph. 29*, 4 (2010), Article 128. 1, 2, 3, 6, 7

[LYvdPG12] LIU L., YIN K., VAN DE PANNE M., GUO B.: Terrain runner: control, parameterization, composition, and planning for highly dynamic motions. *ACM Trans. Graph. 31*, 6 (2012), Article 154. 1, 2, 5

[LYWG13] LIU L., YIN K., WANG B., GUO. B.: Simulation and control of skeleton-driven soft body characters. *ACM Trans. Graph. 32*, 6 (2013), Article 215. 1, 3

[MdLH10] MORDATCH I., DE LASA M., HERTZMANN A.: Robust physics-based locomotion using low-dimensional planning. *ACM Trans. Graph. 29*, 4 (July 2010), 71:1–71:8. 2

[MLPP09] MUICO U., LEE Y., POPOVIĆ J., POPOVIĆ Z.: Contact-aware nonlinear control of dynamic characters. *ACM Trans. Graph. 28*, 3 (2009). 2

[MPS06] MCCANN J., POLLARD N. S., SRINIVASA S.: Physics-based motion retiming. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, Eurographics Association. 8

[MTP12] MORDATCH I., TODOROV E., POPOVIĆ Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph. 31*, 4 (July 2012), 43:1–43:8. 2

[MZS09] MACCHIETTO A., ZORDAN V., SHELTON C. R.: Momentum control for balance. *ACM Trans. Graph. 28*, 3 (2009). 2

[NF02] NEFF M., FIUME E.: Modeling tension and relaxation for computer animation. In *SCA* (2002), ACM, pp. 81–88. 8

[PK09] PETERS J., KOBER J.: Using reward-weighted imitation for robot reinforcement learning. In *Adaptive Dynamic Programming and Reinforcement Learning* (March 2009), pp. 226–232. 2

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3 (2004), 514–521. 2

[SKL07] SOK K. W., KIM M., LEE J.: Simulating biped behaviors from human motion data. *ACM Trans. Graph. 26*, 3 (2007), Article 107. 2

[TGLT14] TAN J., GU Y., LIU C. K., TURK G.: Learning bicycle stunts. *ACM Trans. Graph. 33*, 4 (July 2014), 50:1–50:12. 2

[TK05] TAK S., KO H.-S.: A physically-based motion retargeting filter. *ACM Trans. Graph. 24*, 1 (Jan. 2005), 98–117. 8

[WFH09] WANG J. M., FLEET D. J., HERTZMANN A.: Optimizing walking controllers. *ACM Trans. Graph. 28*, 5 (2009), Article 168. 2

[WK88] WITKIN A., KASS M.: Spacetime constraints. In *Proceedings of SIGGRAPH 1988* (1988), pp. 159–168. 2

[WP09] WAMPLER K., POPOVIĆ Z.: Optimal gait and form for animal locomotion. *ACM Trans. Graph. 28*, 3 (2009). 2

[YLvdP07] YIN K., LOKEN K., VAN DE PANNE M.: SIMBICON: Simple biped locomotion control. *ACM Trans. Graph. 26*, 3 (2007), Article 105. 2

[YN03] YAMANE K., NAKAMURA Y.: Dynamics filter - concept and implementation of online motion generator for human figuresr. *IEEE Transactions on Robotics and Automation 19*, 3 (2003), 421–432. 8