

Diverse Motion Variations for Physics-based Character Animation

Shailen Agrawal Shuo Shen Michiel van de Panne
University of British Columbia

Abstract

We present an optimization framework for generating diverse variations of physics-based character motions. This allows for the automatic synthesis of rich variations in style for simulated jumps, flips, and walks. While well-posed motion optimization problems result in a single optimal motion, we explore using underconstrained motion descriptions and then optimizing for diversity. As input, the method takes a parameterized controller for a successful motion instance, a set of constraints that should be preserved, and a pairwise distance metric between motions. An offline optimization then produces a highly diverse set of motion styles for the same task. We demonstrate results for a variety of 2D and 3D physics-based motions and show that this approach can generate compelling new motions.

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

Keywords: human simulation, control

1 Introduction

Physics-based motions are difficult to author. As such, optimization has been a powerful tool in designing the motion of physics-based characters. Given an objective function and an initial guess at a motion or its controls, a variety of numerical optimization methods can be used to refine the motion control (or the motion itself) until the process converges to a result that maximizes (or minimizes) the objective function. In this context, a well-posed optimization problem should have a well-defined maximum (or minimum) and have an objective function that accurately captures the user’s intent. Objective functions that minimize control effort are a common way of defining such an optimization problem. However, this also unfortunately eliminates the possibility of seeing the rich non-optimal variations in style that are commonly seen in human motions and which help give life to animated motions.

In this paper we propose to work with *underconstrained* motion specifications such as “do a backflip and land in this region” and to then automatically generate highly-diverse motion variations that satisfy the given specifications. This enables the exploration of “the space of back flips” and, similarly, the spaces of possibilities for other types of underconstrained dynamic motions. We loosely refer to these spaces of possible motions as *motion null spaces*.

We expect that the proposed method can be applied in a variety of scenarios. It can enhance the suite of applications that physics-based synthesis methods currently target. It can allow animators to reduce the amount of manual iteration required to design a motion.

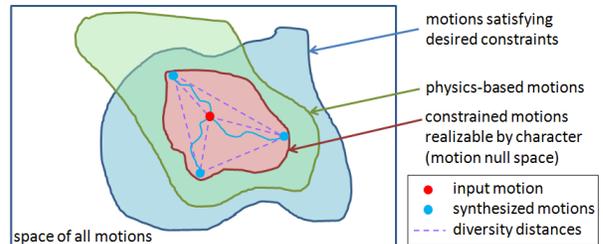


Figure 1: Abstract view of diversity optimization.

Informal discussions with animators reveal that when a new motion is encountered quite some time is spent exploring for a motion that has the “right” look. Lastly, diverse motion synthesis can be used in situations when motion variation becomes important, i.e., for crowds or non-repetitive motions for an individual.

Our primary contribution is a framework that uses *diversity optimization* to allow for the synthesis of sets of simulated motions that span the many possible ways in which an *underconstrained motion* can be achieved. As key elements of this framework, we propose: (1) an objective function specifically tailored to diversity optimization for physics-based character animation; (2) the use of round-robin covariance matrix adaptation (CMA) as an effective optimization strategy; and (3) the use of known pose *similarity metrics* as being equally suitable as *distance metrics* for measuring motion diversity.

An abstract view of the diversity optimization problem is shown in Figure 1. The motion null space is described by the subspace of motions that satisfy physics, satisfy the motion constraints, and are realizable by the character and its underlying controller parameterization. In this paper we do not recover the full motion null space, but rather discover a set of maximally-diverse examples that help define its extent. Given an input motion, shown in red, the goal is to develop synthesized variations that are as distant as possible from each other and the input motion. The straight lines in Figure 1 illustrate the relevant pairwise distances for the synthesis of three new diverse motion variations. The degree of achievable motion variation is constrained by factors such as joint limits, torque limits, and the particular choice of parameterization of the underlying control. We call these *emergent constraints* because it is difficult to know in advance how these implicit and explicit limitations will end up shaping the space of realizable motions for a given task.

2 Related Work

Physics-based character control can be achieved in a variety of ways. Many methods are based on tracking a reference motion based on motion capture data, as seen in many recent methods: [Sok et al. 2007; Yin et al. 2007; Muico et al. 2009; Lee et al. 2010; Kwon and Hodgins 2010; Yin et al. 2007; Ye and Liu 2010; Da Silva et al. 2008; Liu et al. 2010; Coros et al. 2011] A second approach is to develop appropriate objective functions without relying on captured data and to then synthesize motions using online or offline optimization. Representative examples of this approach

include [Liu and Popović 2002; Macchietto et al. 2009; Wang et al. 2009; de Lasa et al. 2010; Wu and Popović 2010; Borno et al. 2013]. A third approach involves users in authoring the shape of the controlled motion, e.g., [Liu and Popović 2002; Yin et al. 2007; Coros et al. 2010; Nunes et al. 2012]. In practice, methods often combine multiple aspects of these three possible sources of motion. The majority of methods are focused on locomotion tasks and result in a single motion style.

Numerous **kinematic models for motion style and variation** have been proposed. Procedural noise [Perlin 1995] or learned models of noise [Lau et al. 2009] can be added to kinematic character motions to achieve motion variety and realism. Many statistical models have been developed to learn motion styles or to estimate style and content in a separable fashion, e.g., [Brand and Hertzmann 2000; Hsu et al. 2005; Wang et al. 2007]. However, these methods create kinematic models of existing styles and are limited in their ability to create new and strikingly different styles of motion.

Variations of **physics-based motions** have also been explored. The passive motion of objects can be shaped so that objects roll and bounce to achieve given goals [Chenney and Forsyth 2000]. The stochastic nature of the search allows for a variety of solutions to be generated for the same problem. Natural sources of variation such as motor noise and environmental conditions can be used to achieve a degree of variation in walking gaits [Wang et al. 2010]. Another approach to generating motion variety is to provide tools that enable the user to efficiently sift for desirable solutions among many automatically-generated variations. This is applied to passive physics-based simulations in [Twigg and James 2007] and to a control system for a 24-DOF monopod hopper dog in the Design Galleries (DG) approach [Marks et al. 1997]. This latter example is closest in spirit to the problem we wish to tackle. The motion of the hopping dog is shaped by 7 time-varying sinusoids that modulate the forward velocity, hopping height, and the pose of the ears, tail, and neck. The amplitude, offsets, and frequencies of these sinusoids are then explored to produce motion variations. Our method works on significantly more complex human and robot figures performing a much wider range of more constrained motion tasks.

Diversity optimization is a problem of general interest in the context of planning and AI [Hebrard et al. 2005; Coman and Muñoz-Avila 2011; Srivastava et al. 2007; Ursem 2002]. The problem is often posed as finding a diverse set of k configurations of a discrete set of parameters such that a given goal, i.e., a target price range of a product, is satisfied. A common diversity metric is to take the sum of a pairwise distance metric over all possible pairs of the k configurations. More recently, the selection of diverse solutions has also been addressed for problems such as designing furniture layouts [Merrell et al. 2011], where solutions generated using Monte Carlo sampling can be used to produced a diversified list using a maximal marginal relevance criterion.

We draw inspiration from the above work in order to develop a method that can propose novel motion styles for physics-based motion tasks. In order to ensure that a good exploration of the space of realizable motions is achieved and is available to the user, we seek to find the *most diverse* set of motion styles. In doing so, the optimization naturally finds novel forms of coordinated full body motion as well as exposing the emergent constraints that eventually serve to limit the extent of possible motion variation. Our problem is such that the cost of finding new samples that satisfy the task constraints is a challenging problem as the solution set becomes more distant from the input motion. As such, we develop an algorithm that takes an integrated approach to maintaining task constraints and producing motion diversity. This stands in contrast to approaches that generally assume a lightweight sample-generation process [Marks et al. 1997; Wang et al. 2010; Merrell et al. 2011].

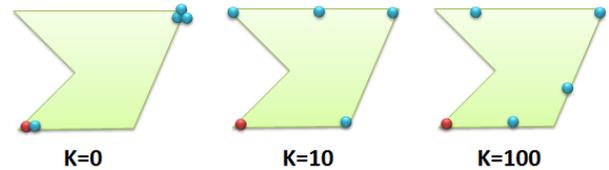


Figure 2: Effect of K on diversity optimization in a 2D domain with a sum-of-squared-distances metric. All 4 synthesized solution points begin at the reference point shown in red.

3 Diversity Optimization

The goal of the diversity optimization is to generate a set of N synthesized motions which are as different as possible from each other while still satisfying desired constraints. This is captured by maximizing the following diversity objective:

$$D = \sum_{i=0}^N \left(-C(M_i) + \sum_{j=0}^N (d(M_i, M_j) + Kd_{\min}(M_i)) \right) \quad (1)$$

where $\{M_i\}$ denotes the set of N motions, $C(M_i)$ is a positively valued constraint function, $d(M_i, M_j)$ is a distance measure between two motions and $d_{\min}(M_i)$ is the minimal distance of all pairwise distances involving motion M_i . K is a weighting constant that we discuss in further detail. The motions themselves are parameterized via their underlying controller according to $M_i = M(P_i)$, where P_i are the controller’s free parameters (§5).

The first term of the objective function, given by the constraint function, evaluates to zero when the motion constraints are satisfied. This “null space” corresponds to a subspace in the parameter space that contains the realizable motion styles that are of interest. Without loss of generality, we assume that the constraint is initially satisfied for M_0 , the reference motion that is used as a starting point, i.e., $C(M_0) = 0$. If this is not true, then $C(M_0)$ is used as an objective function in order to first satisfy this condition. The $d(M_i, M_j)$ term encourages diversity through a maximization of the sum of pairwise distances between the motions. However, by itself this can yield clusters of motions that have negligible distances within the cluster. The $Kd_{\min}(M_i)$ term addresses this by encouraging *dispersion* between all motion pairs, as measured by the minimum pairwise distance for any given motion.

Figure 2 illustrates the effect of assigning different values to K , the weight assigned to the d_{\min} term, for an example where squared Euclidean distances are used as the distance measure. The solutions shown are generated from a stochastic optimization (CMA [Hansen 2006]). $K = 0$ results in undesirable clusters of points located at extremal points of the feasible domain because the benefit of the large inter-cluster distances outweighs the benefit of increasing the intra-cluster distances. As shown by the $K = 100$ result, large values of K can resolve the clustering problem but result in CMA converging to solutions that fail to find extremal points in the domain due to an insufficient reward for diversity. It should be noted that increasing K does not always imply a larger value for d_{\min} , as can be also be seen from the figure. An appropriate middle ground is to choose a value for K that encourages both diversity and dispersion. We use $K = 10$ for all our results.

We develop a *round-robin CMA* algorithm for maximizing D , developed on top of a standard CMA implementation. This applies one generation of CMA optimization to each of the N motions, in turn, before moving on to the next generation of the optimization for each of the motions. During the optimization of motion

variation i , the other motions are held fixed. Optimization is performed using 500-1000 CMA generations with 16 samples per generation. CMA is terminated when the maximum generation count is reached (1000) or when the diversity objective advances by less than a given ϵ . Good results are typically obtained within the first 500 generations.

Each of the N CMA optimizations maintains its own state information, including a mean and covariance matrix associated with its samples. Because the objective function is constructed over mutual distance metrics, the results of optimizing a given motion M_i will also inadvertently change the objective function values seen by the other motion variations. An assumption of the round-robin optimization process is that these changes will be sufficiently small to avoid adverse behavior. In practice, we have not encountered adverse behavior. An alternative strategy is to add motion variations one at a time, each time optimizing only the most recently added motion. In practice, we found that this *progressive addition* strategy performed worse than the round-robin method (§6). All our results are thus computed using round-robin CMA optimization. One other potential strategy is to treat all N motions as a single large optimization problem having $N \times |P|$ parameters, where $|P|$ denotes the cardinality of the set of free parameters to be optimized for each motion. This strategy results in an N -fold increase of the number of optimization parameters and thus it does not scale well.

4 Motion Distance Metrics

The pairwise distance metric provides the foundation for the measurement of the diversity of the solution set. We investigate three possible choices: (1) hands + feet (HF); (2) mass distance (MD); and (3) weighted joint angles (WJA). These choices correspond to pose similarity metrics that are commonly used to compute good transition points between motions. The distance metrics between corresponding poses q and q' are defined according to:

$$d_{\text{HF}}(q, q') = \sum_{i \in \text{hands, feet}} \|\mathbf{x}_i - \mathbf{x}'_i\|^2$$

$$d_{\text{MD}}(q, q') = \sum_i m_i \|\mathbf{x}_i - \mathbf{x}'_i\|^2$$

$$d_{\text{WJA}}(q, q') = \sum_i w_i (\theta_i - \theta'_i)^2$$

where \mathbf{x}_i and \mathbf{x}'_i correspond to the locations of the center of mass of link i for pose q and q' , respectively, after the character's centers of mass have been aligned for the two poses. θ refers to the character's joint angles and root angle. The MD distance metric has been previously used as a similarity metric [Lamouret et al. 1996; Kovar et al. 2002]. We use the WJA weights suggested in a previous study on pose similarity metrics [Wang and Bodenheimer 2003].

The distance between two motions is computed as a sum of the pose distances for a number of corresponding sample points on the pair of motions to be compared. Meaningful correspondences are obtained by using the phase-based structure of the motions (§5). Within each phase, correspondences are established using the normalized time, \hat{t} , where $\hat{t} = (t - t_{\text{start}})/(t_{\text{end}} - t_{\text{start}})$. For walking motions, we use 10 samples for each of four motion phases. For jumping motions, we use 10 samples for each of the two airborne motion phases and do not sample the other phases because the most interesting variations should occur during airborne segments.

Interesting motion variations can be found using all three distance metrics. Many of the results given in the rest of the paper and the accompanying video use the mass-distance metric because of its convenient parameter-free nature.

5 Models and Control Parameterization

2D human model: The planar human figures have 17 links and are simulated using Box2D [Box2D] or Vortex [CMLabs]. After taking symmetry into account, there are 10 joints that require independent control. The character has a mass of 80 kg and a height of 160 cm. High-valued torque limits are used so as to allow the character to perform highly dynamic motions that can make for compelling animations: 300 Nm for the hips and upper-waist, 200 Nm for the knees, ankles, lower-waist, and shoulders, and 50 Nm for the remaining joints. All joints are controlled using proportional-derivative (PD) controllers with gains that are manually set to appropriate values for the input motions. The feet, pelvis, and upper torso are assigned target orientations in the world coordinate frame during stance phase. In all other cases, joints have target orientations specified in local coordinates, i.e., with respect to their proximal (parent) link.

3D human model: This model has 15 links and 24 internal degrees of freedom. The shoulders, hips, waist, ankles, neck and wrist joints are modeled as U-joints, while the elbows and knees are modeled as hinge joints. The character has a mass of 80 kg and height of 180 cm. Torque limits are 200 Nm for the hips and waist, 150 Nm for the knees and ankles, 50 Nm for the neck, and 100 Nm for the shoulders, elbows, and wrists. Joints are driven using PD controllers. For walking gaits we track the pelvis, swing-leg femur, and swing-leg foot orientation in a world-aligned coordinate frame [Yin et al. 2007]. During jumps, we similarly track the desired pelvis orientation in world space during stance phases. Otherwise, all joints track joint-local target angles. The forward dynamics is simulated using Vortex [CMLabs].

3D bird-robot walker: The model has 10 links and 14 internal degrees of freedom. The middle pair of joints on each leg are modeled as hinge joints and the remaining joints are U-joints. The character weighs 70 kg and is 2.9 m tall. The control setup is analogous to that of the 3D human model.

Controller phases: Motions are the product of a parameterized controller and forward dynamics simulation. The control is broken into multiple phases as shown in Figure 3 for jumping motions and walking motions. Transitions between phases occur after a phase-specific time, denoted by Δt_i or upon a transition event, i.e., becoming airborne, reaching peak height, or making ground contact. Each phase i has an associated target pose, Ω_i . Within phase i , the target pose is interpolated linearly over time between Ω_{i-1} and Ω_i using the known transition time. For phases that terminate upon transition events, a transition time is estimated using a simple prediction mechanism. An exception is the *extend* phase of the jump, which directly transitions to its target pose in order to rapidly create upward momentum. A balance controller is used during the start and end phases of jumps and is implemented using virtual forces applied to the center of mass using a Jacobian transpose model. Lastly, a simple landing controller helps to accelerate the motion optimization by providing intelligent foot placement. This is only implemented for our 2D character. In the falling phase of jumps, the feet are servoed to the target landing position where the COM is predicted to pass through the ground, with an additional offset that is a linear function of the horizontal velocity.

Optimization free parameters: The principal free parameters in the diversity optimization are the target poses for each phase of the controller. For 3D human jumps, all sagittal joint angles are included, with the exception of the wrists and neck, yielding 49 optimization parameters. For 3D human walks, all joints are included (sagittal and non-sagittal) except for the neck joints, for a total of 52 optimization parameters. For 2D jumps, all joints are included, with the exception of the neck. Additional parameters include the

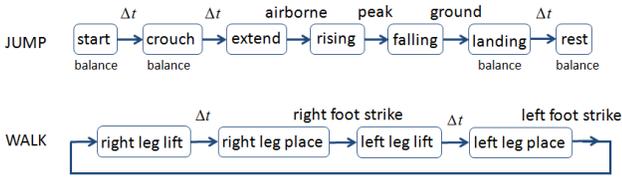


Figure 3: Motion control phases.

transition times and phase-specific k_p, k_d gains for the hips, knees, and ankles, for a total of 101 parameters.

Constraints: The constraint function C is modeled using a penalty that evaluates to zero when the constraints are satisfied and applies a quadratically rising penalty outside of this region. The constraint violations should be weighted sufficiently to rapidly dominate the diversity objective term once constraints are violated. In practice we find that provided it is sufficiently large, the specific choice of weight is not important given that the optimization method (CMA) is invariant with respect to order preserving transformations of the objective function value. Like a barrier method, the constraint provides a gradient for the optimization to follow to achieve accommodation of the constraints, while following the diversity gradient when constraints are satisfied. Obtaining meaningful results requires that a suitable constraint null space exists. We revisit this issue in the discussion of results.

6 Results

We explore the creation of a set of four diverse motion variations for nine constrained motion tasks. The results are illustrated in Figure 4 and in the accompanying video. The differences between some of the motion variations are best observed when viewing them at half speed (see A/V controls panel in QuickTime Player). For a given motion task, we use M_n to refer to the motion variations, where n denotes the row number. The top row, M_0 , corresponds to the input motion and the remaining rows, M_1 – M_4 , correspond to the four synthesized motion variations. We refer to the individual image columns using the letters A–F. Images within a column are chosen to correspond to a particular motion phase, which is more meaningful than a time-based correspondence.

Four maximally-diverse motion variations can be synthesized in 30 minutes using 16 threads on an 8 core Xeon machine using an unoptimized multithreaded implementation with Vortex and with 40% CPU usage. All the 3D examples are simulated using Vortex, as are the 2D jump-high and jump-forward shown in the accompanying video. A time step of 120 Hz is achieved using the “position-locked” PD-control available in Vortex. The remainder of our 2D simulations are implemented using our original multithreaded Box2D implementation with 16 threads on an 8-core Xeon machine. These run at 3900 Hz because of the explicit integration of the PD-controllers. Computing 4 variations of an input motion thus requires 1–4 hours of computation for these motions.

6.1 Motion Tasks

Backflip: We first create diverse variations for a standing backflip with a landing constraint. The landing constraint that requires the center of mass of the foot to end within a 30 cm landing box. The constraint penalty is zero for landing within this box. Constraint costs outside of this region are modeled according to $C(M) = K\|d\|^2$, where $K = 5000$ and d is the Euclidean distance by which the constraint is violated. Falls are given a large constant penalty, as is a failure of the root link to achieve a near-

360° rotation. The constraint violation penalty and falling penalty are also modeled in a similar way for the remaining motions, unless stated otherwise. An input motion is manually authored using the given phase structure. The landing constraint is placed to match the landing location of the input motion.

Figure 4(a) illustrates four motions created from a single diversity-optimization run and uses the WJA distance metric. M_1 and M_2 show significant variations in takeoff pose (C), overall height (D), and landing pose (F). M_3 learns to assume a pike pose during the flight phase. While some of the motions exhibit qualities that might not be capable of being matched by an athlete, these are also the qualities that make the motion appealing. If desired, the limits of human musculature and joints can be modeled with more fidelity in order to achieve more subdued results. We have also successfully experimented with backflip variations without a landing constraint, which leads to back flips that also vary in their horizontal distance traveled.

Jump forward: We compute variations for a forward jump with a landing-box constraint, as shown in Figure 4(b) and using the mass distance metric. A manually authored controller is first developed that is capable of a small in-place jump. This is then optimized using CMA, with the landing-box constraint penalty serving as an objective function. The resulting motion is the input motion for the diversity optimization. The same manually authored controller also serves as a starting point for the development of the input motions for all the other jump motions. M_1 and M_3 produce jump variations with different styles of back arches, while M_2 realizes a jump with considerable forward pitch of the torso. M_4 bears a resemblance to M_0 but considerably differs in phases C–E.

Jump onto: This motion is constrained to land on top of the box, with the landing constraint implementing a small safety margin. The input motion is obtained by optimizing the small in-place jump to first jump forward onto a box of height $h = 0$. The motion is then adapted to a box height of $h = 0.5$ m using a continuation-based optimization [Yin et al. 2008]. The degree of available style variation decreases as the box height increases and so we therefore stop at a moderate box height for this example. The weighted joint-angle distance metric is used to create the four motion variations shown in Figure 4(c). M_1 and M_2 create jump styles that perform pikes, each with its own style of arm motion. M_3 performs a tuck jump with a forward-pitched torso. M_4 develops a jump with a backwards arch.

Jump over: A jump over an obstacle is implemented using an obstacle-clearance constraint and a landing constraint that enforces a minimal jump length. The input motion is developed in an analogous fashion to the Jump Onto motion. The small in-place jump is first optimized to meet the required jump length. The motion is then adapted to clear an obstacle height of $h = 0.55$ m using continuation-based optimization. Four diverse motion variations are synthesized using the weighted joint angle distance metric and are illustrated in Figure 4(d).

Jump high: An in-place jumping motion is developed using two constraints: a minimal-height constraint for the peak-COM height and a landing constraint. The input motion is created by optimizing the small in-place jump to satisfy the given constraints. Four diverse motion styles are produced using the mass distance metric and are shown in Figure 4(e). The resulting variations include two different styles with backwards arches (M_1, M_2) and two jumps, M_3 and M_4 , that perform pikes of varying extents in conjunction with different types of coordinated arm movements.



Figure 4: Results for 2D motions (a,b,c,d,e) and 3D motions (f,g,h,i). For each task we show the input motion, M_0 , on the first row and the four synthesized variations, M_1 – M_4 , on the remaining rows.

Backwards jump: The backwards jump for the 3D human model requires landing within a 0.25 m of a target point that is located 0.8 m behind the character. The input motion comes from a manually authored jump that is then optimized to satisfy the landing constraint. Four diverse style variations are synthesized using the mass distance metric and are shown in Figure 4(f). The resulting dynamic motions exhibit a wide range of mid-air poses.

Robot jump: This jump uses a landing constraint located 4 m ahead of the starting location with a radius of 0.75 m. An additional upper body rotation constraint is used to prevent excessive forward pitch. A manually authored small jump is optimized to satisfy the landing constraint which then serves as the input motion. Four diverse styles are produced using the mass distance metric (see Figure 4(g)). The styles achieve a wide range of mid-air poses.

Human and Robot silly walks: A diverse range of exaggerated dynamic walks can be achieved, making it easy to create physics-based walks that are reminiscent of Monty Python’s “Ministry of Silly Walks” [YouTube]. A step-length constraint is applied in the diversity optimization results shown in Figure 4(h,i). Each step requires the character to achieve a step length of $(1.0 \pm 0.5, 0.0 \pm 0.5)$ m for the human and $(1.2 \pm 0.3, 0.0 \pm 0.3)$ m for the robot, where (x, y) represents the forward and lateral length of the step. An upper body orientation constraint requires that the pelvis and torso remain within 15° of the vertical. The character is evaluated from a static position corresponding to the pose demanded by the first phase of the controller and is simulated for 30 steps. The first 10 steps are used to allow the character to attain a limit cycle behavior. The constraints and motion diversity are then evaluated over the last 20 steps. If the character falls during the first step, a penalty of 50,000 is applied, with the penalty linearly decreasing to 0 for falling after the 30th step. This provides a suitable gradient for the optimization to exploit. Four variations are synthesized using the mass-distance metric. The resulting styles are highly diverse and dynamic in nature. The robot silly walks have particularly interesting styles: M_1 is a tip-toe walk; M_2 has a serious and purposeful tone; M_3 is an asymmetric loping walk; and M_4 yields a chin-in-the-air proud style of walk.

6.2 Control over Magnitude of Motion Variation

We provide two methods for control over the magnitude of motion variation. The first takes advantage of the progressive nature of the CMA optimization, which provides a solution path from the initial motion to each of the maximally diverse motions. We parameterize this path according to the underlying distance metric, and allow the user to explore intermediate points along these paths. A second method is to place intuitive constraints on the controller. We experiment with two different strength models for motions, a default strength model and a second ‘supernatural’ strength model that increases the joint PD constants and torque limits by a factor of 1.5–3. These two methods are illustrated in Figure 5 and the accompanying video. A third method would be to constrain the motion variations to be within some desired ϵ of a reference motion in terms of energy expenditure or distance metric. We leave the exploration of this idea as future work.

The set of motion variations can be further shaped through additional motion constraints, the creation of an initial motion that more closely resembles the desired class of motions, and refining the choice of free parameters according to the degrees of freedom where the variation is desired.

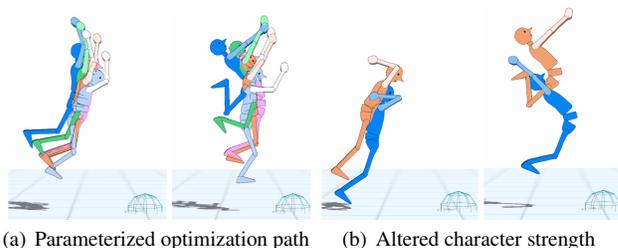


Figure 5: Two methods for controlling the degree of diversity. (a) The figures in light blue, orange, pink, green, and blue show results at points that are 0, 25, 50, 75, and 100% along the optimization path. (b) The orange figures is significantly stronger than the blue figure.

6.3 Performance

We conducted a simple test to compare the simultaneous optimization of N motion variations, as performed by the round-robin CMA, against the progressive addition strategy (§3) that only optimizes the most recently added motion until it is as diverse as possible. This is tested for $N = 4$ on the backflip problem. Using the results from 5 runs we obtain $D_{\min} = 41.8$, $\bar{D} = 54.4$, $D_{\max} = 64.1$ for the one-at-a-time results, and $D_{\min} = 71.2$, $\bar{D} = 83.5$, $D_{\max} = 103.7$ for the $N = 4$ simultaneous optimizations. The results show that the worst-case simultaneous optimization still significantly outperforms the best one-at-a-time optimization. Another naive approach would be to generate many motion variations that satisfy the task constraint and then simply retain the N variations that are most diverse. However, the problem domain is such that there is no efficient way to generate motion variations that satisfy the task constraint, and particularly not for large motion variations that satisfy the task constraint. The method presented in this paper is motivated in part by the desire to develop such a motion null space.

We use a set of four motion variations, i.e., $N = 4$, to provide a consistent picture of the types of motion variations that are generated. In general, repeating the stochastic optimization may produce additional motion variations. While computing the required distance metrics is $O(N^2)$, the dominant cost lies with the simulations used to evaluate the CMA samples. As such, the overall diversity optimization is effectively $O(N)$. Our experiments with $N = 10$ show that this does produce additional distinctive motion variations, although the mean distance between motion variations does begin to fall.

7 Discussion and Conclusions

The ability to automatically synthesize diverse styles of physics-based motions provides for a new type of ‘imagination amplification’ for animation. The synthesized motions continue to diverge from each other until they are limited by some combination of joint limits, torque limits, ability to recover balance, and the built-in constraints of the controller parameterization. The interplay of these limiting factors is such that it is difficult for an animator to pre-conceive of how they might shape the space of possible motions. Our work provides a new tool for exploring the space of possible motions. While some exploration of possible motions is also achievable with current optimization methods by manually adding or reweighting objective function terms, we argue that this is neither convenient nor principled if the objective function must be specifically tailored to each motion. We further show that pose similarity metrics drawn from previous work can also be used as distance met-

rics for achieving diverse motion variations.

The method may fail to produce expected styles of motion variation for several reasons. The solution may not converge to a global maximum. The solution may be strongly multimodal with considerable distances between modes, in which case CMA may have trouble finding distant modes. We have observed one case of a flip being discovered as a new means for performing a high jump. Undesirable motion variations may also occur if the parameterization of the controller is ill suited to producing natural human motions. As a result, the design of the underlying controller and the choice of initial motion will also influence the final result. The quality of our results for common motions such as walking depend on the viewer's expectations of the character. The walk variations produced for the robot are plausible while those produced for the human model are physically plausible and entertaining but not natural.

Care needs to be taken to avoid overconstraining the motion. A motion that is tightly constrained, such as hitting a particular keyframe at a given point in time, will have a limited subspace within which to optimize for diversity and our method would likely have trouble meeting such specific constraints. The optimization problem also benefits from the use of feedback-based control that is built into the motion parameterization, such as the balancing controller or the use of a SIMBICON-style feedback loop in our 3D walking examples. These feedback structures allow the optimization to make faster progress and also result in a more robust simulated motion. An alternate approach would be to investigate the use of trajectory-based optimization methods for such problems, instead of the control-based method that we currently employ.

An interesting direction for future work would be to explore the design of a multidimensional motion null space with the help of our diversity results. We currently only support the exploration of continuously parameterized solutions along the optimization paths resulting from the CMA solutions, i.e., the result shown in Figure 5(a). It may be possible to use the current results to define a large subspace in which arbitrary convex combinations of control parameters, P_j , yield a continuous multidimensional motion null space that contains significantly more variations than that observed along the optimization paths alone. Effort metrics can be incorporated into the system by performing an epsilon-optimal search. Another exciting possibility would be to use a video-to-controller system [Vondrak et al. 2012] in order to quickly and automatically create the input motion and its controller from a direct demonstration. Diversity optimization could then be used to immediately create feasible style variations that are constrained to have a similar outcome.

References

BORNO, M. A., DE LASA, M., AND HERTZMANN, A. 2013. Trajectory optimization for full-body movements with complex contacts. *IEEE Trans. Visualization and Computer Graphics*. in press.

BOX2D. Box2d v2.2.1, <http://box2d.org/>.

BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *Proc. ACM SIGGRAPH*, ACM, 183–192.

CHENNEY, S., AND FORSYTH, D. 2000. Sampling plausible solutions to multi-body constraint problems. In *Proc. ACM SIGGRAPH*, ACM, 219–228.

CMLABS. Vortex simulator, v5.1.

COMAN, A., AND MUÑOZ-AVILA, H. 2011. Generating diverse plans using quantitative and qualitative plan distance metrics. In *Proc. of AAAI*, 946–951.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Transactions on Graphics* 29, 4, Article 130.

COROS, S., KARPATY, A., JONES, B., REVERET, L., AND VAN DE PANNE, M. 2011. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics (TOG)* 30, 4, 59.

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Simulation of human motion data using short-horizon model-predictive control. In *Computer Graphics Forum*, vol. 27, Wiley Online Library, 371–380.

DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-Based Locomotion Controllers. *ACM Transactions on Graphics* 29, 3.

HANSEN, N. 2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, Eds. Springer, 75–102.

HEBRARD, E., HNIC, B., O SULLIVAN, B., AND WALSH, T. 2005. Finding diverse and similar solutions in constraint programming. In *Proc. AAAI 2005*, vol. 20, 372–377.

HSU, E., PULLI, K., AND POPOVIĆ, J. 2005. Style translation for human motion. In *ACM Transactions on Graphics (TOG)*, vol. 24, ACM, 1082–1089.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics*, 473–482.

KWON, T., AND HODGINS, J. K. 2010. Control systems for human running using an inverted pendulum model and a reference motion capture sequence. *The ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA 2010)*.

LAMOURET, A., VAN DE PANNE, M., ET AL. 1996. Motion synthesis by example. In *Eurographics Workshop on Computer Animation and Simulation*, 199–212.

LAU, M., BAR-JOSEPH, Z., AND KUFFNER, J. 2009. Modeling spatial and temporal variation in motion data. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 171.

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

LIU, C., AND POPOVIĆ, Z. 2002. Synthesis of complex dynamic character motion from simple animations. In *ACM Transactions on Graphics (TOG)*, vol. 21, ACM, 408–416.

LIU, L., YIN, K., VAN DE PANNE, M., SHAO, T., AND XU, W. 2010. Sampling-based contact-rich motion control. *ACM Transactions on Graphics* 29, 4, Article 128.

MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. 2009. Momentum control for balance. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 80.

MARKS, J., ANDALMAN, B., BEARDSLEY, P., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUMML, W., ET AL. 1997. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*, ACM Press/Addison-Wesley Publishing Co., 389–400.

- MERRELL, P., SCHKUFZA, E., LI, Z., AGRAWALA, M., AND KOLTUN, V. 2011. Interactive furniture layout using interior design guidelines. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 87.
- MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z. 2009. Contact-aware nonlinear control of dynamic characters. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 81.
- NUNES, R., KRY, P., AND ZORDAN, V. 2012. Using natural vibrations to guide control for locomotion. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 87–94.
- PERLIN, K. 1995. Real time responsive animation with personality. *IEEE Trans. on Visualization and Computer Graphics 1*, 1, 5–15.
- SOK, K. W., KIM, M., AND LEE, J. 2007. Simulating biped behaviors from human motion data. *ACM Trans. Graph.* 26 (July).
- SRIVASTAVA, B., KAMBHAMPATI, S., NGUYEN, T., DO, M., GEREVINI, A., AND SERINA, I. 2007. Domain independent approaches for finding diverse plans. In *Proc. IJCAI*, Morgan Kaufmann Publishers Inc., 2016–2022.
- TWIGG, C., AND JAMES, D. 2007. Many-worlds browsing for control of multibody dynamics. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 14.
- URSEM, R. 2002. Diversity-guided evolutionary algorithms. *Parallel Problem Solving from Nature PPSN VII*, 462–471.
- VONDRAK, M., SIGAL, L., HODGINS, J., AND JENKINS, O. 2012. Video-based 3d motion capture through biped control. *ACM Transactions on Graphics (TOG)* 31, 4, 27.
- WANG, J., AND BODENHEIMER, B. 2003. An evaluation of a cost metric for selecting transitions between motion segments. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 232–238.
- WANG, J., FLEET, D., AND HERTZMANN, A. 2007. Multifactor gaussian process models for style-content separation. In *Proceedings of the 24th international conference on Machine learning*, ACM, 975–982.
- WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2009. Optimizing walking controllers. *ACM Trans. Graph.* 28 (December), 168:1–168:8.
- WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2010. Optimizing walking controllers for uncertain inputs and environments. *ACM Trans. Graph.* 29 (July), 73:1–73:8.
- WU, J.-C., AND POPOVIĆ, Z. 2010. Terrain-adaptive bipedal locomotion control. *ACM Transactions on Graphics* 29, 4 (Jul.), 72:1–72:10.
- YE, Y., AND LIU, C. K. 2010. Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph.* 29 (July), 74:1–74:9.
- YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. *ACM Trans. Graph.* 26, 3, Article 105.
- YIN, K., COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2008. Continuation methods for adapting simulated skills. *ACM Trans. Graph.* 27, 3.
- YOUTUBE. Monty python, "ministry of silly walks". <https://www.youtube.com/watch?v=9Z1BUg1E6Hc>.