Soft Walks: Real-Time, Two-Ways Interaction between a Character and Loose Grounds

Chloé Paliard[1], Eduardo Alvarado[2], Damien Rohmer[2], Marie-Paule Cani[2]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris [2] LIX, Ecole Polytechnique/CNRS, Institut Polytechnique de Paris

Figure 1: *New layers added to a walking character in a game engine allow to automatically deform a variety of loose terrains and grass, using simple procedural parameters. Changes of gait that are consistent with this evolving environment are generated in real-time.*

Abstract

When walking on loose terrains, possibly covered with vegetation, the ground and grass should deform, but the character's gait should also change accordingly. We propose a method for modeling such two-ways interactions in real-time. We first complement a layered character model by a high-level controller, which uses position and angular velocity inputs to improve dynamic oscillations when walking on various slopes. Secondly, at a refined level, the feet are set to locally deform the ground and surrounding vegetation using efficient procedural functions, while the character's response to such deformations is computed through adapted inverse kinematics. While simple to set up, our method is generic enough to adapt to any character morphology. Moreover, its ability to generate in real time, consistent gaits on a variety of loose grounds of arbitrary slope, possibly covered with grass, makes it an interesting solution to enhance films and games.

1. Introduction

Real-time animation of characters walking on loose grounds such as sand, snow, mud or soil, possibly covered with grass, is a key element in video games and virtual-reality applications involving natural environments. Modeling plausible walking motions on such terrains requires two-ways interactions. Indeed, not only should the character's feet deform the loose material of the ground, but this deformation should, in return, dynamically impact the walking style, since both balance and motion IKs are to be adapted to the evolving terrain.

While recent work based on deep and/or reinforcement learning allows to animate character motion on terrains with a variety of rigid slopes and obstacles, real-time character animation on loose grounds and deformable vegetation remains a challenge. Indeed, capturing enough learning data would be difficult, while optimizing motion-controllers would require costly training sessions where all collisions and interactions between the character and the environment are fully simulated. These simulations remain notably out of reach when loose material and soft interactions are involved.

This work takes a different path, in proposing a very simple and efficient, yet versatile, method to modify a character's walking style, through two-ways interactions between specific layers of its motion controller and the environment, the latter being consistently deformed through efficient, procedural functions. Since it achieves real-time performances, our method is well suited to game engines such as Unity (Unity Technologies), which we use for illustration.

Our main contribution is a new, multi-layer animation modifier for the character (Sec. 3) which applies at different levels of details while reusing and extending standard game-engine principles, namely combination of approximated physics-based dynamics, pre-set motions and inverse kinematics (IK). We first introduce, at a high level, a global orientation controller allowing the character to dynamically tilt and improve balance by targeting an equilibrium state that depends on the nature and slope of the terrain. Second, at

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

DOI: 10.2312/egs.20211019

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

a finer scale, we model two-ways interactions between the character's feet and the environment by using simple distance-based rules used to both dynamically deform the environment and adapt feet placement and character walking style in real-time. Lastly, we show how such layered control can be used to enhance standard game engines in different scenarios, such as walking on sand, soil, mud, snow and grass with different slopes (Sec. 4). In each case, a specific deformation is applied to the environment, which consistently impacts the walking style. The method runs in real-time, and opens the way to extensions to other loose environments, as well as refinement in adding more layers to the character's motion controller.

2. Related Work

Modeling plausible character motion is a complex problem, which couples behavioral intents with physics-based equilibrium constraints. Fully kinematic models were explored in early work [SM01] leading to real-time adaptation of walking cycles to uneven terrains, at the price of lacking dynamic response to terrain changes. Coupling preset gaits with dynamic events was handled through the help of animation controllers [YLvdP07] that convert differences between the actual pose and the prescribed one as torques, which are then fed into a physically-based simulation computing dynamic state changes. Coupled with reinforcement-learning [PALvdP18] and/or adapted reward functions [KLP20], these approaches can model a wide range of plausible complex motions up to acrobatic effects [WGH20]. These motions automatically adapt to their simulated environment, at the price of their high computational cost and indirect control, and thus cannot be directly applied to loose deforming terrains in standard game engines. In reverse, gathering and processing large prerecorded motion databases allow for the use of deep-learning methods [HKS17] which, once learned, can achieve real-time performance. Still, the need of an extensive motion data-base would not scale well to the variety of loose environments targeted in this work.

Closer to our goals, simple dynamic models applying forces to the center of mass to control the general balance of a character, seen as an inverted pendulum [KH10], allow a good high-level trade-off between efficiency and plausibility. Mitake et al. [MAA*09] proposed a high level representation of an oscillating proxy volume, triggering pre-recorded detailed animations (walking, running, falling). This work achieves real-time animation but is restricted to pre-set animations on flat grounds. Other works attempt to exploit to a greater degree the environmental impact on the dynamic model. Bermudez et al. [BTZZ18] proposed a method for applying drag forces to a character walking in a fluid and adapting its motion to the medium. Similarly to ours, this approach rely on high level controller-based model. However, this previous work only take into account the influence of the environment, ie. the fluid, toward the character's gait, but not the coupled two-ways interactions between them as in this work.

3. A layered model for character motion

Our method takes as input an animated character model from a standard game engine, ie. a mesh geometry, a rigged skeleton with preset animations, an IK system applied to feet bones, on top of a simple proxy-geometry used for collision processing, and a global dynamic behavior computed in simulating a single rigid body model expressed by a center of mass and an inertia tensor. This model allows the character to walk on flat and sloppy terrains, but does not allow to dynamically adapt its gait to varying ground slopes, leading to a robotic-looking motion that lacks dynamic movement. Furthermore, it does not model any two-ways interaction with the environment, therefore restricting the animation to rigid grounds.

The aforementioned elements can be seen as a layered model for character motion-control, where each layer acts at a different level of detail. They span from global control for the rigid body that controls the position and orientation of the coarse geometric collider, to local control for the IK, which positions, computed using dynamic ray-casting, allow for accurate feet positioning on the ground. In this work, we enhance the two layers we just mentioned, by firstly adding an extra swing-control term to the global rigid body model, and secondly by dynamically adapting feet IKs, enabling them to trigger deformations of the surrounding ground and grass. Taken together, these contributions result in consistent interactions between a walking character and an arbitrary, natural-looking environment.

3.1. A high level controller for tilting and swinging motion

The first layer of our model is a single global controller used to generate a torque applied to the rigid-body. This controller allows the character to self-stabilize on a steep terrain, while adding a dynamic swinging behavior conveying the notion of the effort developed by the character to remain stable and move forward. While controllers are usually expressed as a difference between the current angles and some objective angles for specific limbs, our high-level controller aims at stabilizing the general character, which does not have a well defined notion of angle in itself. Instead, we propose to use the more relevant notion of center of mass, and in particular its projection along the gravity direction to the support polygon of the character.

Since we aim at modeling a tilting motion in the velocity direction, we simplify the representation to a 2D-planar problem, where the plane is defined by the gravity direction, the current character velocity, and is passing through its center of mass. The support polygon is dynamically computed as the segment defined by the projection of the feet positions onto the ground in this plane. The objective projection of the center of mass p_{com}^{target} is thus expressed as the middle of this segment (see Fig. 2-left). The torque magnitude is finally computed as a proportional derivative controller:

$$\tau = \alpha\,(p_{com}^{target} - p_{com}) \cdot u + \beta\,\dot{\theta}\,, \qquad (1)$$

where p_{com} and p_{com}^{target} are respectively the coordinates of the current and the target projections of the center of mass onto the ground, u is the unit vector along the terrain slope, and $\dot{\theta}$ is the angular velocity of the rigid body around the axis orthogonal to the plane. The control parameter α is used to tune the freedom of the character to tilt out of its ideal equilibrium position, while β sets how much of dynamic swinging is allowed. The values of these parameters can easily be adapted to the nature of the environment as explained in Sec. 4. As shown on Fig. 2-middle and right, this new high-level controller automatically adapts to various slopes and character morphologies, without requiring any parameter tuning. To ensure the accuracy of foot positioning and limit any sliding artifact, a small offset is added on the character with the PD controller to match the IK-objective of the animation.

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

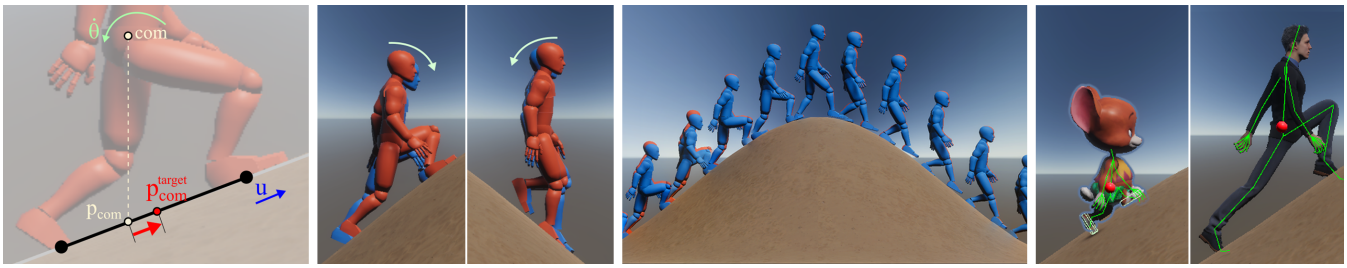

Figure 2: *Left: The torque magnitude depends on the distance between the middle of the support polygon p_{com}^{target} and the projection of the character's center of mass p_{com} on the terrain, as well as on the current angular velocity $\dot{\theta}$. Middle: Comparison between characters without (red) and with (blue) our torque-based controller. The blue character exhibits an increased swinging motion, which dynamically tilts it forward or backward, depending on the slope. Right: The controller is robust to changes of character morphology, since the oscillating motion automatically adapts when the center of mass (red sphere) is positioned closer to or further from the ground.*

3.2. Two-ways interactions between feet and loose grounds

The second, new control layer models both the deformation of the environment and its reciprocal action on the character's gait. To accurately compute the contact between the foot and the ground, which can be deformed dynamically, our method relies on the use of ray-casting, emitted vertically from the foot to the ground. Once the contact point is computed, it serves as an IK key to the associated leg. This approach is complemented by mechanisms modeling the loose nature of the ground, and possibly some soft volumetric layer (eg. grass) above it.

Interaction with the ground. The ground is represented as a height-field grid on which feet dynamically imprint. We inspire from Sumner et al. [SOH99] for procedurally computing footprints: The geometry of the next support foot hitting the ground carves it locally, while removed material is added on the side of the footprint. We use the following three scalar parameters to adapt the deformation to different types t of grounds:

The depth coefficient d_t defines the height subtracted in one frame to each grid-cell of the terrain in collision with the feet. Thus, smaller values correspond to harder terrains.

The compression coefficient c_t describes the amount of compression the ground material can bear, and therefore controls the volume of material to be moved to the edge of the print (eg. mud is compressible, as some water will flow out of the footprint under pressure, while dry sand is not). In practice, the height of the grid-cells surrounding the feet is increased by c_t.

The smoothness coefficient s_t tunes the aspect of ground deformation. It corresponds to the magnitude of a Gaussian filter applied to the deformed height-field. A high value models material such as dry sand, where fine grains lead to a smooth appearance.

Interaction with loose surrounding elements. In addition to loose grounds, surrounding elements defined in a volumetric layer above it, such as short or dense vegetation, act as loose material and should be impacted, and change the character motion in return.

The action of the feet on all geometric elements in such layer is handled using a procedural deformation, function of the distance to the walking foot: We choose to apply a translation t with linearly decreasing magnitude around the foot position p_f, computed at a point p as:

$$t(p) = \text{clamp}\left(t_{max} - \gamma\|p - p_f\|\right) \frac{(p - p_f)_{down}}{\|p - p_f\|}, \qquad (2)$$

where t_{max} and γ are respectively control parameters for the magnitude and influence area of the translation, and v_{down} denotes a modified vector that always points downward, computed as: $v_{down} = (v_x, -|v_y|, v_z)$, where y is the vertical direction.

Figure 3: *We use a moving platform (right) acting as a fake target ground for the foot, but suddenly moving down when hit, to model the presence of grass and its ability to be crushed.*

Secondly, surrounding elements should directly affect characters' walking styles. One may observe on footage of natural gaits within short, dense vegetation, that humans tend to raise their knees higher and land more abruptly when reaching a short and dense vegetation layer. Indeed, walkers cannot accurately perceive the actual distance from their feet to the ground, but only roughly estimate it. They thus typically target higher positions, while the vegetation will suddenly crush under their feet. We capture this using virtual platforms placed under the feet, and modeling the perceived ground (see Fig. 3). These platforms, serving as targets for the IK system, are procedurally controlled depending on the height of the soft layer (eg. vegetation). During the upward foot motion phase, the associated platform progressively rises and uplifts the foot. During downward motion, it is set to a specific, fixed height. Lastly, when the foot reaches it, the platform either falls down quickly or just disappears (depending on the targeted material for the above-ground layer), leading to a sudden adjustment of the character feet onto the actual ground.

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

4. Application to various terrains

We implemented our method within the Unity Game Engine[†], running in real-time on standard laptops, and used it to model interactions between a walking character and a variety of terrains, namely dry sand, soil, mud and snow. We considered 10×10 meters terrains with a maximum height of 10 meters and character sizes ranging from 1 to 1.9 meters. Height fields were typically sampled using 512^2 points.

Footprints computed on terrains without vegetation are illustrated in Fig. 4. Controller parameters in Eq. (1) were set to $\alpha = 30$, and $\beta = 6$. The variety of prints was achieved by tuning the depth, compression, and smoothness parameters described in Section 3.2. Dry sand was set to show limited depth changes, while exhibiting non-compressible behavior and large smoothing. Soil was set to smaller depth but higher compression than sand. Mud and snow were set to allow deeper carving, prints on mud being set to be sharper that those on snow. The resulting instantaneous change of terrain height – controlled through the depth coefficient – has an immediate effect on the character's gait, as shown on the joint video.

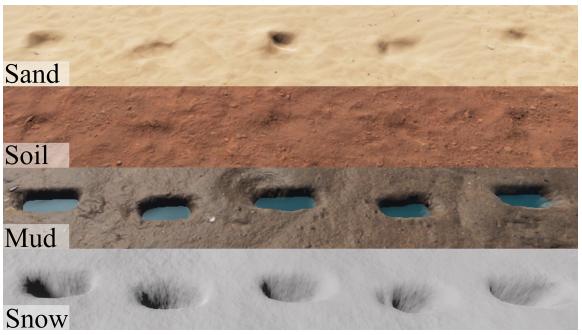

Figure 4: *Footprints on different terrains.*

The grass was represented as billboards (see Figure 3 and the accompanying video), and its deformation was implemented in Unity using shader-graphs to allow efficient space deformations. We considered two typical grass heights : small (0.5 meters) and medium (0.9 meters). The size of the vegetation influences the maximal height of the platform set at one third of the vegetation's height. We also forced the character to swing less in the case of higher vegetation, to model friction. To this end, we adapted the controller parameter β to range linearly from 6 (no vegetation) to 4 (medium vegetation).

More examples combining different terrains, slopes and vegetation are shown in Fig. 1 and in the supplemental video. In all situations shown, the character automatically adapts its gait while dynamically interacting with the surrounding, loose environment.

5. Limitations and Future Work

This work shows that simple layers acting at different levels of motion controllers can enrich real-time character animation, and handle two-ways interactions with natural environments. Simple and efficient, our method achieves consistent changes of the environment and the character's gait.

Among limitations, sand footprints do not account for slope, leading to symmetrical shapes even on steep dunes where sandslides should be taking place. Secondly, our handling of vegetation is limited, billboards being unable to capture realistic deformations. We also fail to handle different natures and densities of vegetation, such as plastic deformations for broken rods. Lastly, our prototype implementation only models straight walks on small terrains. Extending balance control to orthogonal motion directions would allow the character to turn, while using adaptive, local subdivision of the terrain would allow to freely navigate the character on larger and visually richer terrains.

An interesting avenue for future work would be the addition of new control layers to other parts of the character to improve both its global and local adaptations to the environment. We are particularly interested in extending the approach to other part of the character such as the motion of his arms through high vegetation to clear bushes or protecting its head. Extension toward more accurate impact on the environment could be explored through learning local deformation that could be obtained from simulation or real data, while reinforcement learning could be used to improve the response on some layer of the character such as its feet placement. Lastly, our method was specifically designed for walking motions on quite stable terrains.Generalizing control and interaction layers could enable to handle more complex cases, such as a character or any other creature, running, jumping or climbing a scree with rolling stones.

References

[BTZZ18] BERMUDEZ L., TESSENDORF J., ZIMMERMANN D., ZORDAN V.: Real-time locomotion with character-fluid interactions. *MIG 18* (2018). 2

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Trans. Graph. 36*, 4 (2017). 2

[KH10] KWON T., HODGINS J.: Control Systems for Human Running using an Inverted Pendulum Model and a Reference Motion Capture Sequence. *SCA* (2010). 2

[KLP20] KWON T., LEE Y., PANNE M. V. D.: Fast and flexible multilegged locomotion using learned centroidal dynamics. *ACM Trans. Graph.* (2020). 2

[MAA*09] MITAKE H., ASANO K., AOKI T., SALVATI M., SATO M., HASEGAWA S.: Physics-driven multi dimensional keyframe animation for artist-directable interactive character. *Comput. Graph. Forum 28*, 2 (2009). 2

[PALvdP18] PENG X. B., ABBEEL P., LEVINE S., VAN DE PANNE M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph. 37* (2018). 2

[SM01] SUN H. C., METAXAS D. N.: Automating gait generation. In *ACM SIGGRAPH* (2001). 2

[SOH99] SUMNER R. W., O'BRIEN J. F., HODGINS J. K.: Animating Sand, Mud, and Snow. *Comput. Graph. Forum* (1999). 3

[WGH20] WON J., GOPINATH D., HODGINS J.: A Scalable Approach to Control Diverse Behaviors for Physically Simulated Characters. *ACM Trans. Graph.* (2020). 2

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

[†] https://unity.com

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.