THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Character contact re-positioning under large environment deformation

Citation for published version:
Tonneau, S, Ali Al-Ashqar, R, Pettré, J, Komura, T & Mansard, N 2016, 'Character contact re-positioning under large environment deformation' Computer Graphics Forum, vol. 35, no. 2, pp. 127-138. DOI: 10.1111/cgf.12817

Digital Object Identifier (DOI):
10.1111/cgf.12817

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Graphics Forum

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 05. Apr. 2019

EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Character contact re-positioning under large environment deformation

Steve Tonneau[1†], Rami Ali Al-Ashqar[2†], Julien Pettré[3], Taku Komura[2], Nicolas Mansard[1]

[1]LAAS-CNRS / UPS
[2]University of Edinburgh
[3]Inria Rennes

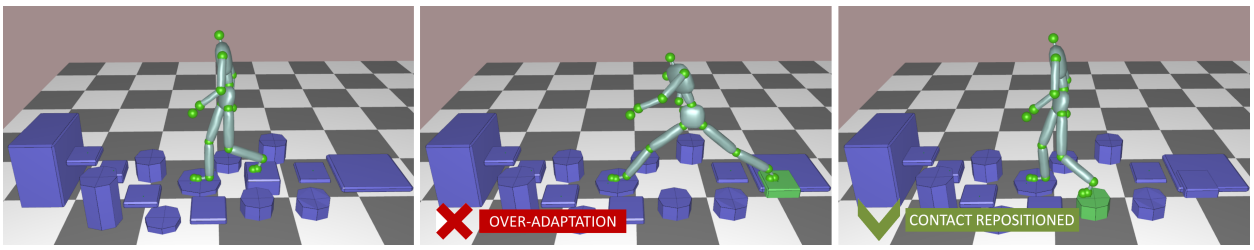


Figure 1: *Left: Reference motion. Middle: The user moves away one of the step stones. The motion is over-adapted based on the relationship-descriptors technique. Right: the contact is re-positioned, resulting in a more plausible motion.*

Abstract

Character animation based on motion capture provides intrinsically plausible results, but lacks the flexibility of procedural methods. Motion editing methods partially address this limitation by adapting the animation to small deformations of the environment. We extend one such method, the so-called relationship descriptors, to tackle the issue of motion editing under large environment deformations. Large deformations often result in joint limits violation, loss of balance, or collisions. Our method handles these situations by automatically detecting and re-positioning invalidated contacts. The new contact configurations are chosen to preserve the mechanical properties of the original contacts in order to provide plausible support phases. When it is not possible to find an equivalent contact, a procedural animation is generated and blended with the original motion. Thanks to an optimization scheme, the resulting motions are continuous and preserve the style of the reference motions. The method is fully interactive and enables the motion to be adapted on-line even in case of large changes of the environment. We demonstrate our method on several challenging scenarios, proving its immediate application to 3D animation softwares and video games.

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

1. Introduction

One of the long standing objectives of computer animation is to synthesize plausible motions for characters interacting with complex environments. Plausibility is usually obtained by animating characters using motion capture data. Unfortunately, this process lacks flexibility, as the content of a motion capture is fixed. To overcome this limitation, several motion editing methods have been proposed. They perform geometric adaptations of the reference motion to small variations of the environment. In general however, large variations result in unnatural deformations, because of collisions, joint limit violations, or loss of balance in the resulting animation.

Our objective is to tackle these large deformation scenarios, and thus to extend the validity domain of motion adaptation techniques. The hard issue on which we focus is the re-positioning of contacts that become invalidated by the environment edition, while preserving the style of the reference motion. For instance consider a motion where a character stands up from a chair by creating a contact with his hands and the armchairs. How to play the motion in a context

† Steve Tonneau and Rami Ali Al-Ashqar are joint first authors

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

where the chair has no armchairs? In most cases, simply removing the contacts is not sufficient because of balance issues. The invalid contacts either have to be re-positioned, or the motion adapted to respect balance constraints.

Our approach to the problem is incremental. We start from a state of the art motion editing method [AAKC13], and use it to adapt the motion to small deformations of the environment. While the motion is adapted, we automatically determine whether the deformation is acceptable regarding collision, balance, and joint limits, by considering each limb in contact individually. This is done efficiently thanks to a new method we propose, based on the reachable workspace of the considered limb.

If one criterion is not verified, the contact phase is invalidated, and a new contact location is computed. The search for new contacts is based on a fast heuristic-driven search among possible surfaces of the environment. Among the candidates, the new contact must verify the mentioned conditions. Regarding force exertion, the new contact must also contribute to the overall motion in the same manner that the original contact did. This is verified with a new heuristic, the force actuation profile. In a last step, if contact re-positioning fails, we use a motion planner to adapt the root trajectory to the environment. The planner is biased to preserve the style of the original motion and to minimize contact invalidation.

These adaptations are smoothly blended within the original motion, thanks to a local optimization method that minimizes the differences between the original and adapted motion.

Our contribution can be summarized as follows:

- An automatic and efficient method to detect unnatural motion adaptations. The method is generic and can be applied with any motion editing technique.
- An interactive contact re-positioning method that handles large deformations of the environment.
- A low-dimensional sampling based method able adapt a root trajectory to large deformations when contact re-positioning fails.
- A novel optimization scheme that preserve the continuity and style of the adapted motion.

In the remainder of the paper, we first situate our contribution with respect to the related work in Section 2. We then provide an overview of the method in Section 3. One section is then dedicated to each step of the method (Section 4 to Section 7). Section 8 presents our experimental results, and provides support for a discussion on the method in Section 9. Section 10 concludes the paper with a discussion on current applications and future works.

2. Related work

This paper presents a procedural approach to adapt continuously an existing motion captured example. Our work is thus related to several approaches: physically-based procedural animation, data-driven motion synthesis and motion adaptation.

Physically-based Procedural Animation These methods aim at controlling a character using physical models. They often rely on finite state machines and PD control to guide the motion [Tho91, vdPL95, YLvdP07]. Designing such controllers for complex environments is not trivial: the required transition rules and states are less obvious than for cyclic walking patterns, and much more numerous. To address the limitation, methods based on reinforcement learning have been proposed, though they suffer from a huge amount of memory usage and precomputation. [PBvdP15]. Contact invariant control [MTP12], on the other hand, does not require designing a state machine. A set of contact variables is considered within a constrained optimization problem, such that contact phases are automatically discovered along the generation of the motion. However, the resulting motion is not dynamically consistent, is sensitive to local minima, and requires several minutes of computation. Interactive motion synthesis can be achieved using sequential Monte Carlo approaches [HET*14] or particle belief propagation [HRL15]. Although they provide an exciting direction of research, as of today the synthesized motions lack naturalness, especially for motions involving close environment interactions.

Data-driven motion synthesis Using motion capture data is the most straight-forward way to synthesize plausible human motion. Simply replaying the captured motion lacks flexibility. A number of techniques such as motion blending [WP95, RBC98] and inverse kinematics [BB04] are proposed to increase the flexibility of the captured data. These methods enable adapting a motion by interpolating or warping motions to meet user-defined constraints. Motion graphs couple such approaches with a data structure where nodes are motion clips, connected if a continuous transition exists between them. This enables synthesizing complex motions that result from the concatenation of the traversed nodes [KGP02, LCR*02, AF02].

Various data-driven approaches based on motion capture data are proposed to control characters in novel environments. Methods based on PRMs [CLS03, PLS03] are effective to evaluate the areas where the characters can move into and find the set of available movements at such areas. Kim et al. [KHKL09] propose an approach to spatial-temporally edit the motion while preserving the context of interactions, by inserting/removing gait cycles when the trajectories are stretched/shortened, and curving the trajectories by Laplacian editing when it is bent. Safonova and Hodgins [SH07] parameterize the movements by interpolating the motions in the same category, and compute the sequence of movements by dynamic programming. Min and Chai [MC12] introduce such a framework in the probabilistic domain. Levine et al. [LLKP11] extend the idea of sample-based planning to the temporal domain, and compute the optimal series of movements even taking into account dynamic obstacles. Lee et al. [LCL06] parameterize and associates movements to spatial building blocks called motion patches, and produce large scale scenes by stitching them together.

Motion Adaptation of Close Interactions Highly-constrained cases may however result in failures of data-driven methods. For instance, when entering a car, maintaining the contacts and avoiding collisions are contradictory objectives due to the proximity of obstacles, thus hard to satisfy when the geometry is modified. Such scenarios may require to use importance-based techniques [SLSG01] to take into account the relations of all adjacent objects. Ho et al. [HKT10] represent the context by producing a

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

volumetric graph between the character's joints and the environment. The motion is adapted using Laplacian mesh deformation [SCOL*04], that preserves the local geometry of the interaction. Relationship descriptors by Al-Ashqar et al. [AAKC13] propose a frame-by-frame adaptation scheme, that fits the requirements of real-time applications. The limitation of these methods is reached when the collision avoidance and contact maintenance constraints become incompatible (due to a large deformation), so that contact re-positioning becomes necessary. Generating automatically a replacement contact is possible [TPM14], but the existing method does not consider motion continuity, nor the style of the motion.

State of the art analysis Generating plausible motions that involve close interactions is a recognized open issue. Optimization-based techniques that take into account physics requires off-line computation in this context. Data-driven methods give plausible results thanks to local deformations of a reference natural motion. However the amplitude of adaptations that can be applied to the reference motion is a crucial question to decide which method to use. In close interaction scenarios, especially when concavities exist, this amplitude is low due to the limited space available for the character. In this context large deformations require contact re-positioning, because collisions prevent contact maintenance, or because the contact geometry has been deleted! For existing methods to be applied in the context, two important issues have to be solved: the automatic detection of these failure cases, and the automatic contact re-positioning that solves them while preserving motion continuity. In this paper we demonstrate this feasibility: we extend the work of [AAKC13], by addressing these two issues.

3. Overview and organization

Our method consists in one preprocessing phase, and one interactive phase. The workflow is illustrated in Fig. 2. The interactive phase is designed to provide an appropriate answer to the environment modifications: large adaptations are only applied if small adaptations fail first. The method is fully automatic.

During preprocessing, a static analysis of the original animation is performed. It extracts the features that describe the interactions with the environment: information regarding the proximity between each body part and obstacles is stored (relationship descriptors); contact phases are identified, and characterized regarding their contribution to the motion (contact descriptors).

At runtime, if an object of the environment is deformed, the character motion is adapted using relationship descriptors. The importance of the adaptation depends on the proximity between the character's bodies and the object in the original motion, as proposed by Al-Ashqar et al. [AAKC13].

Our contribution extends this method. In parallel, efficient routines check a set of kinematic and dynamic conditions that ensure that the adaptation remains plausible. When a condition is violated, the motion is invalidated, and locally replaced with an equivalent one. Specifically, new contact configurations are chosen to provide force actuation capabilities similar to the original ones. If contact re-positioning fails, the last step consists in locally re-positioning the root trajectory of the character. This is achieved using a random sampling scheme, biased with a "reachability condition". This condition favors root positions close to the original ones, that avoid collisions, and preserve balance. In any case, the continuity of the motion is always preserved thanks to a local optimization scheme.

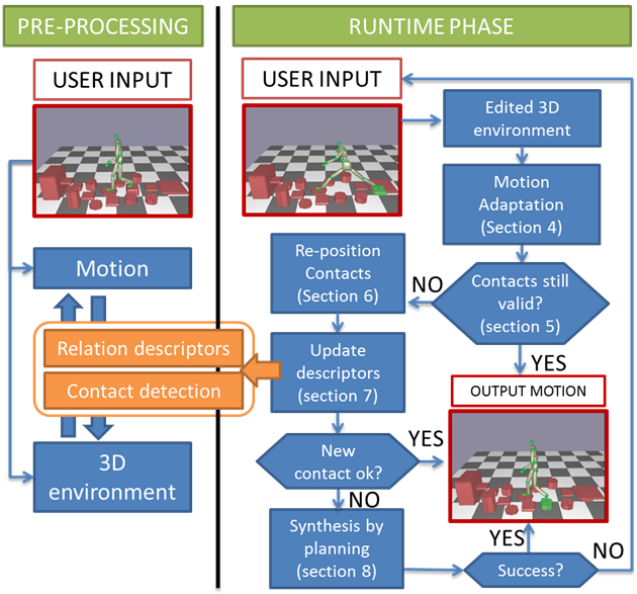


Figure 2: *Work-flow of the method.* ***Left****: an automatic preprocessing analysis of the motion allows to describe relationship and contact descriptors.* ***Right****: Depending on the deformation, the interactive phase can take up to three steps, from motion deformation to larger contact and body re-positioning.*

4. Preprocessing of the reference motion

Adapting a reference motion to environment deformation requires addressing two sub-problems. First, at the kinematic level, to identify the relationship between the character joint positions and the environment, to understand how the deformation of the environment affects the motion. Then at the physical level, to identify the contact phases regarding their length and their contribution to the motion. This information allows to re-position the contacts while preserving the two properties. We compute these two descriptors (relationship descriptors, contact descriptors) automatically from the reference motion as follows.

4.1. Relationship Descriptors

Representation based on relationship descriptors [AAKC13] allows a smooth adaptation of an animation to new contexts, when the geometry (of the environment or of the character) is deformed. We quickly recall it here for completeness. Rather than representing the configuration of the avatar by a configuration vector $\mathbf{q}$, the body positions are described by their relative translations from a static set of points called descriptor points (Fig. 3). The descriptor points are sampled on the surface of the environment. The closer the body is to the surface, the denser the sampling is going to be.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

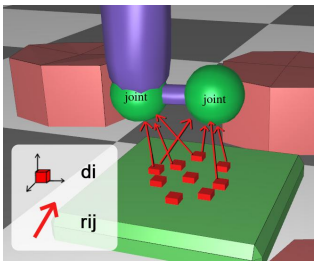


Figure 3: *The joint positions are described as a weighted sum of relative vectors originating from descriptor points embedded in the environment.*

In this representation, the joint position $\mathbf{p_j}$ of joint j is represented by the following equation:

$$\mathbf{p}_j = \sum_i^N w_{i,j}(\mathbf{d}_i + \mathbf{r}_{i,j}) \tag{1}$$

where $\mathbf{d_i}$ is the position of the i-th descriptor point, $\mathbf{r}_{i,j}$ is the vector between the i-th descriptor point and the j-th joint, and $w_{i,j}$ is the weight of each vector whose value fades out according to the length of $\mathbf{r}_{i,j}$ and its angle with respect to the normal vector at the descriptor point. See the App. 1 for the calculation of the weights.

Rather than storing the initial trajectory as a collection of joint positions $\mathbf{p_j}$, we store, for each body, a reference to the descriptor points $\mathbf{d}_i$ along with the coordinates of $\mathbf{r}_{i,j}$ and the weights $w_{i,j}$, because it encodes the spatial relations between the body and the environment, i.e., the context of the scene.

4.2. Contact Descriptors

We first identify the contact phases of the motion. We then compute a force actuation profile to characterize each contact phase.

4.2.1. Definitions

A character R has l limbs $R^k, 1 \leq k \leq l$ attached to the root (i.e. for a humanoid, 2 arms and 2 legs).
$\mathbf{q}^0 \in SE(3)$ is the position and orientation of the root of R;
$\mathbf{q}^k$ is the set of internal joint angle values of a limb of R^k;
$\mathbf{p}^k(\mathbf{q}) = \mathbf{p}^k(\mathbf{q}^0, \mathbf{q}^k)$ is the world position of the k-th end-effector;
$\mathbf{p}^k(\mathbf{q}^k) = \mathbf{p}^k(0_{SE3}, \mathbf{q}^k)$ is the same position in the root frame;
The value of a variable for a frame s is noted with the subscript $_s$.

4.2.2. Contact phases

The detection of a contact might be difficult because of the noise in the input data (i.e. the end-effector is not exactly in contact with the environment or not perfectly static). We thus assume that a contact occurs at frame s if both the distance to the environment and the velocity of the effector are close to zero. This means that the following condition holds:

$$d_s < \varepsilon_0 \quad \text{and} \quad v_s \leq 2\varepsilon_1 \tag{2}$$

where d_s is the minimal Euclidian distance between the end-effector body and the environment, $v_s = ||\mathbf{p}^k(\mathbf{q}^k_{s+1}) - \mathbf{p}^k(\mathbf{q}^k_{s-1})||$ is the instantaneous velocity of the end-effector and ε_0, ε_1 are user defined threshold variables (empirically set to 0.01 and 0.001).

4.2.3. Force actuation profile of a contact phase

Each contact is used by the character to exert a force that contributes to the motion. To re-position a contact, we must allow the character to exert a similar force, otherwise the motion is not plausible. To characterize this contribution, one possibility is to explicitly reconstruct the contact forces with inverse dynamics. However, such methods are very sensitive to noise (they rely on an estimation of the joint acceleration) and are not observable (e.g. when both feet are on the ground, it is not possible to decide what are the respective tangential forces at each foot). We propose an efficient dual approach, that does not compute explicitly the contact forces, but measures the bounds on these forces in any direction.

To compute these bounds, we first follow the steps of Yoshikawa [Yos84]. We then extend its formulation to take into account the quality of the contact surfaces.

For brevity, we omit the k exponents in the equations. The relationship between end-effector and joint velocities is given by:

$$\dot{\mathbf{p}}(\mathbf{q}) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

where $\mathbf{J}(\mathbf{q})$ is the jacobian matrix. For a force vector $\mathbf{f}$ and an equivalent joint torque τ, the mechanical work is defined as:

$$\dot{\mathbf{q}}^T\tau = \dot{\mathbf{p}}^T\mathbf{f} \tag{4}$$

We then consider the unit ball in the joint space:

$$||\tau||^2 \leq 1 \tag{5}$$

and map it to the Cartesian space using Equations 3 and 4:

$$\mathbf{f}^T(\mathbf{J}\mathbf{J}^T)\mathbf{f} \leq 1 \tag{6}$$

Equation 6 defines the so-called force ellipsoid [Yos84]. It describes the set of achievable forces $\mathbf{f}$ subject to the constraint 5. The length of an axis measures the ratio between the joint torques and the resulting force applied along its direction. Thus the longer the axis, the bigger the force along the axis can be.

Three examples of force ellipsoids are given in different configurations in Figure 4. The two first configurations present a similar, almost isotropic ellipsoid (ie. close to a sphere); it is slightly deformed so that the force actuation capabilities are more important in the "forward up" direction (the longest axis). In the right frame, the ellipsoid is highly anisotropic, which indicates that the configuration is close to singularity: high force actuation capability in the "forward up direction", but almost no capability in the other directions. If we want to re-position the first configuration, between the second and third configurations, we claim that the best candidate to replace the third contact configuration is the second one.

We now characterize a contact configuration, given the force ellipsoid and the surface of contact. The resulting "force actuation profile" allows us to compare candidate contact replacements.

Since we do not know in which direction the force is applied at a contact, we consider the three main directions of the force ellipsoid $\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}$. They are given by the normalized Eigen vectors of

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

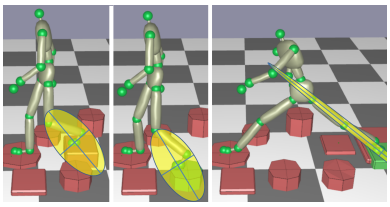


Figure 4: *2D illustration of the force ellipsoid (scale is 0.3). The 2 main axes are drawn with blue lines. The original (left) and re-positioned (middle) contacts present different yet similar force actuation profiles, contrary to the deformed one (right).*

$\mathbf{JJ}^\intercal$. Another constraint on each $\mathbf{u_i}$ to enforce balance is given by Coulomb's non-slipping condition :

$$\mathbf{u}_{ti} \leq \nu_0 \mathbf{n}^\intercal \mathbf{u_i} \tag{7}$$

where $\mathbf{u}_{ti}$ is a tangential component of $\mathbf{u_i}$, and ν_0 and $\mathbf{n}$ are the friction coefficient and the normal of the contact surface. Thus, we define the force actuation profile of a contact configuration $\mathbf{q}^k$ as $[\alpha^{\mathbf{u_1}}, \alpha^{\mathbf{u_2}}, \alpha^{\mathbf{u_3}}]$, with:

$$\alpha^{\mathbf{u_i}} = \alpha(\mathbf{q}^k, \mathbf{u_i}, \nu_0, \mathbf{n}) = \frac{1}{\sigma_i}(\nu_0 \mathbf{n}^\intercal \mathbf{u_i}) \tag{8}$$

where $\sigma_i = \sqrt{\mathbf{u_i}^\intercal (\mathbf{JJ}^\intercal) \mathbf{u_i}}$ is the singular value corresponding to $\mathbf{u_i}$. $\alpha^{\mathbf{u_i}}$ is the EFORT measure function, introduced in [TPM14].

The proposed force actuation profile is thus a low-dimensional characterization of the force actuation capabilities of a contact.

5. On-line motion adaptation through contact re-positioning

The contact and relationship descriptors are used in the on-line phase to adapt the motion to deformations of the environment. Whenever the environment is moderately deformed, the relationship descriptors generate a smooth adaptation. In parallel to the automatic adaptation from the relationship descriptors, efficient tests are performed to check whether the deformation is too large, and a contact is invalidated if needed. In that case an equivalent contact is automatically selected using an efficient contact generator. The descriptors are consequently updated with the new contact information, resulting in a smooth motion adaption.

5.1. Adaptation using relationship descriptors

When a surface is deformed, the descriptor points move rigidly with respect to that surface. To maintain the spatial relationship between a body part and a descriptor point, the associated relative vector must also move rigidly and stay invariant after the deformation. However, keeping the relative vectors between all the joints and the descriptor points invariant is in general impossible. Thus the new joint positions are computed such that they are close to the weighted average of the end-points of the relative vectors originating from the descriptor points:

$$\min_{\mathbf{p}_0 \cdots \mathbf{p}_n} \sum_{i=1}^{n} \left(\mathbf{p}_i - \sum_j w_{i,j} (\mathbf{d}_j + \mathbf{r}_{i,j}) \right)$$

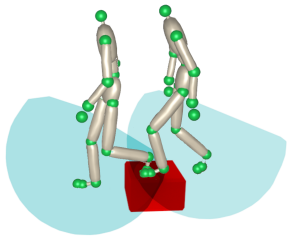


Figure 5: *Reachable workspace of the left foot of the character (blue), at the start and end of a contact phase. The contact can be maintained during the phase because its location belongs to the non null intersection of the workspace at these two different steps.*

$$\text{s.t.} \quad ||\mathbf{p}_i - \mathbf{p}_{i-1}|| = l_i \quad \forall i$$

where l_i is the length of the body segment i. The constraints thus preserves the rigidity of the body parts. The dedicated inverse-kinematics scheme used to solve this minimization problem is proposed in [AAKC13]. For completeness, it is recalled in App. 2.

As a result, a new character posture is obtained that integrates the deformations of the environment.

5.2. Detecting contact invalidation

We propose a novel approach to detect when the deformation of the environment is significant, and thus results in contact invalidation. Outside of collision, contact invalidation can occur for two reasons: joint limits violation and loss of balance. The difficulty of invalidating joint limits comes from the fact that relationship descriptors do not compute the joint values. Rather than performing inverse kinematics to determine these joint values along the motion, we propose a novel, more efficient approach: we verify that the contact location remains in the reachable workspace of the effector during the contact phase.

Joint limit violation. We define the reachable workspace W^k:

$$W^k = \left\{ \mathbf{x} \in \mathbb{R}^3 : \exists \mathbf{q}^k, \mathbf{p}^k(\mathbf{q}^k) = \mathbf{x} \right\} \tag{9}$$

We denote by $W^k(\mathbf{q}^0)$ the volume W^k translated and rotated by the rigid displacement $\mathbf{q}^0$. Figure 5 illustrates W^k for the left leg of the character in two different configurations. A contact position $\mathbf{p_c}$ is out of reach from a limb R_k if

$$\mathbf{p_c} \notin W^k(\mathbf{q}^0) \tag{10}$$

We approximate W^k by off-line sampling 100000 configurations and taking the convex hull of the end-effector positions of this set of configurations.

As an over-estimation of W^k, the convex hull introduces false positives (invalid configurations that are not detected). Under this approximation condition 10 is sufficient, but no longer necessary (Figure 6). From a pragmatic point of view, these false positives correspond to configurations close to the body, and are likely to be invalidated by the collision tests. We performed 100000 tests, where we compared joint invalidation with our approach versus explicit invalidation with inverse kinematics. As a result the convex

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

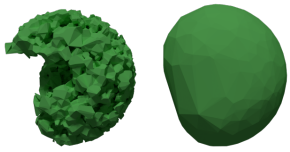


Figure 6: *Two approximations of the range of motion of the right arm. Left: non convex-hull, computed with the powercrust algorithm [ACK01]. Right: convex hull of the reachable workspace.*

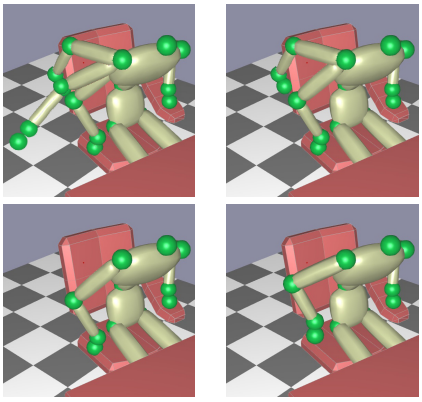


Figure 7: *Contact selection process for the right arm.* ***up-left****: a database of $N = 3$ candidate configurations* ***up-right****: a proximity query eliminates the candidate too far from contact.* ***bottom-left****: the best candidate regarding the force actuation profile is chosen.* ***bottom-right****: After optimization, the contact configuration is modified to be blended in the motion.*

hull introduced in average 4% of false positives. We consider this trade-off as an acceptable cost for efficiency.

Detecting loss of balance. Regarding balance, we use an efficient polytope formulation of the problem [QEMR11]. For a given set of contacts, we are able to define a 6 dimensional polytope, that bounds the admissible center of mass positions and accelerations of the character. We can thus trivially test whether balance is lost upon deformation for the whole contact phase. The algorithm to compute the polytope is provided in [QEMR11].

In summary, we can efficiently determine if a contact phase is invalid: First, testing whether the contact point is reachable, and then test the configurations for collision and balance.

5.3. Contact re-positioning

We now address the issue of re-positioning an invalid contact. The new contact must preserve the properties of the original contact, in terms of duration, force actuation profile and balance. Our novel approach consists in an efficient and accurate heuristic to track the force actuation profile of a contact configuration.

We use a sampling based approach to generate candidate configurations. Details of the method can be found in [TPM14], but we recall the principle for completeness. For each limb 10000 valid configurations are stored off-line in an octree, and indexed by the effector position expressed in the root frame (Figure 7–up-left). Online, the octree is moved to the current root position, and a proximity query is performed to retrieve the configurations close to contact (Figure 7–up-right). Then, the best candidate is chosen according to a user-defined heuristic (Figure 7–bottom-left). In the rest of this section, the candidate list is thus considered to as an input.

Our contribution regarding [TPM14] is the heuristic that we propose here to satisfy the problem constraints, as well as an optimization scheme to preserve motion continuity.

Maximizing the force actuation profile similarity

We consider the issue of ordering the candidates to choose a contact with the same force actuation profile as the original contact. We define n^+ and ν_0^+ as the normal and friction coefficient of the contact candidate, and $\mathbf{q}^{k+}$ as the corresponding configuration. Then we define the score force actuation profile heuristic h as:

$$h_1 = \sum_{i=1}^{3} \alpha^{\mathbf{u_i}} \alpha(\mathbf{q}^k, \mathbf{u_i}, \nu_0^+, \mathbf{n}^+) \tag{11}$$

The interpretation of h_1 is the following: The score associated with a given configuration is higher if the principal axes of its force ellipsoid are aligned with the ones of the original contact, and if their length are similar. The score is also higher if the contact allows to respect the non slipping condition.

Verifying contact duration

If the original contact phase existed for frames s to g, ideally the new contact should last for the same frame window. We enforce this by using the reachable workspace of the limb along the motion (Figure 5). The contact candidate is generated at frame $m = (g - s)/2$, and is located at $\mathbf{p_c}$. We define $s^* \leq m \leq g^*$ as the longest frame sequence verifying:

$$\forall i, s^* \leq i \leq g^*, \mathbf{p}_c \in W^k(\mathbf{q}_i^0) \tag{12}$$

The normalized contact duration score h_2 is then simply defined as:

$$h_2 = (g^* - s^*)/(g - s) \tag{13}$$

Global score function

The global score function is a trade-off between the two heuristics:

$$h = wh_1 + (1 - w)h_2, w \in [0, 1] \tag{14}$$

We empirically set $w = 0.7$.

With h, we obtain an ordered set of candidates. Starting with the best candidate, we verify the balance and collision free conditions sequentially. We return the first candidate that satisfies both.

With the presented scheme, the contact generation is biased towards the most promising contact candidates. This results in physically valid configurations in little computation times. The bias introduced does not discard potential candidates, making the method exhaustive relatively to the set of contact candidates.

If the re-positioning is successful, the contact descriptors are updated accordingly, as well as the relationship descriptors, as described in the following section. The failure case is addressed in Section 7.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

6. Recalculation of the descriptor points

Upon contact repositioning, the global motion must remain smooth and continuous. To preserve this continuity, the relationship descriptors are updated by solving an optimization problem to blend the original and new motions (Figure 7–bottom-right).

Cost function

Let us assume the position of the joints in the original posture $\mathbf{p}_j$ has been updated to $\mathbf{p}_j{}'$ due to repositioning the contacts. Our aim here is to modify the original attributes of the descriptors (describing $\mathbf{p}_j$) so that they now correspond to the new position $\mathbf{p}_j{}'$. We adjust the origin $\mathbf{d}_i$ and orientation $\mathbf{R}_i$ of the descriptors while using the relative vectors $\mathbf{r}_{i,j}$ in the original scene, such that the spatial relations between the environment and the body in the original scene is preserved. The main term of the cost function is then:

$$E_J = \sum_{i,j} \|\mathbf{p}_j{}' - \sum_{i}^{N} w_{i,j}(\mathbf{d}_i + \mathbf{R}_i \mathbf{r}_{i,j})\|^2 \tag{15}$$

In addition to Eq. (15), we consider the following regularization function that represents the difference in translation and rotation between the descriptor and the surrounding ones:

$$E_S = \sum_i \sum_{n_i} (\|\mathbf{d}_i - \mathbf{d}_{n_i}\|^2 + \|\mathbf{R}_i - \mathbf{R}_{n_i}\|_F^2) \tag{16}$$

where n_i is an index for the k-nearest neighbors (here we use $k = 3$), and $\|\cdot\|_F$ is the Frobenius norm. This regularization tends to make the behavior of the collection of descriptors spatially coherent.

Finally, a second regularization term is added to penalize scaling/shearing and large updates from the previous iteration:

$$E_R = \sum_i \|\mathbf{R}_i - \mathbf{R}'_i\|_F^2 \tag{17}$$

where $\mathbf{R}'_i$ is the rotation matrix computed in the previous iteration, that is initialized as an identity matrix. This term alone will not manage to guarantee that $\mathbf{R}_i$ remains a rotation matrix after several iterations; but it is doable by a projection scheme in the minimization scheme, explained below.

We compute the optimal translation and rotation of the relationship descriptors by minimizing the following error function:

$$\min_{\substack{\mathbf{d}_1 \ldots \mathbf{d}_n \\ \mathbf{R}_1 \ldots \mathbf{R}_n}} \quad E_J + w_S E_S + w_R E_R, \tag{18}$$

where w_S and w_R are weights described below.

Minimization scheme

Eq. (18) is a non linear minimization problem that we solve by a dedicated iterative approach. At each iteration of the minimization, the problem is reorganized as a linear least-square problem in the form of:

$$E_{\{J,S,R\}} = \|\mathbf{A}_{\{J,S,R\}}\mathbf{b} - \mathbf{c}_{\{J,S,R\}}\|^2$$

where the decision variable $\mathbf{b}$ is a vector that includes all the elements of $\mathbf{d}_i$ and $\mathbf{R}_i$, i.e., $\mathbf{b} = (\mathbf{d}_1, \mathbf{R}_1^1, \mathbf{R}_1^2, \mathbf{R}_1^3, \ldots, \mathbf{d}_n, \mathbf{R}_n^1, \mathbf{R}_n^2, \mathbf{R}_n^3)^\intercal$, $\mathbf{R}_i^m$ is the m-th row of $\mathbf{R}_i$, and $\mathbf{A}_{\{J,S,R\}}, \mathbf{c}_{\{J,S,R\}}$ are the coefficient matrices and constant vectors from each error function. As a result, we can compute the $\mathbf{b}$ that minimizes an integrated error function $E_J + w_S E_S + w_R E_R$ by

$$\mathbf{b}_o = \arg\min_{\mathbf{b}} E_J + w_S E_S + w_R E_R = (\mathbf{A}^\intercal \mathbf{A})^{-1} \mathbf{A}^\intercal \mathbf{c} \tag{19}$$

where $\mathbf{A} = \left(\mathbf{A}_J^\intercal, w_S \mathbf{A}_S^\intercal, w_R \mathbf{A}_R^\intercal\right)^\intercal$ and $\mathbf{c} = \left(\mathbf{c}_J^\intercal, w_S \mathbf{c}_S^\intercal, w_R \mathbf{c}_R^\intercal\right)^\intercal$.

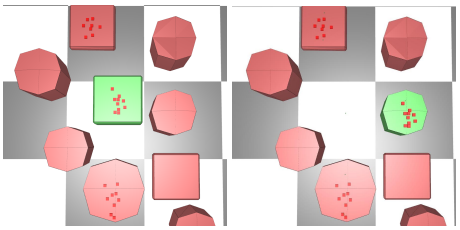


Figure 8: *The distribution of relationship descriptors before and after removing the stepping-stone in the center.*

As mentioned above, the resulting rotation matrix (computed by solving Eq. (19)) includes scaling/shearing. Therefore, we first apply singular value decomposition (SVD) to the computed matrix: $\mathbf{R}_i = \mathbf{U}\Sigma\mathbf{V}$, and then extract the rotation matrix by $\mathbf{R}'_i = \mathbf{U}\mathbf{V}$. This matrix is fed back into Eq. (17) for the next iteration, until E_J is below the threshold.

Regarding the weights of the terms, w_J and w_R are set to 1, while w_S is initially set to 100 and reduced by 1 at each iteration, where the iterations are run 100 times in average. This is to impose more rigidity at the beginning of the optimization and gradually increase the flexibility such that the descriptors can adjust their locations as they converge.

The cost due to rotation difference in Eq.16 and 17 should be ideally measured as rotation difference. This can be done by approximating each rotation by a skew-symmetric matrix, which is a linearization of the rotation matrix [SCOL*04]. We have adopted the current approach due to simplicity of the implementation.

Fig. 8 gives an example where the descriptor points are recomputed after removing a stepping stone in the middle of the environment where the character steps onto. The contact repositioning scheme produces a posture where the character steps on the cylinder at the right. The descriptor points shift over the top of cylinder to correspond to the planned posture. Later displacement of this cylinder will result in a natural adaptation of the planned posture.

7. Root trajectory adaptation

If contact re-positioning fails, in a last step we use a sampling-based motion planner to find a solution despite the large environment changes. Such planners are commonly used to generate motions; however the resulting motions tend to lack realism [LKJ99].

Our objective is to preserve the style of the original motion while regenerating a large part of it to face some drastic changes of the environment. We use a dedicated reachability-based planner [TMP*15] to interactively generate new trajectories for the root $\mathbf{q}^0$. In this work the planner is implemented as a RRT for performance.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

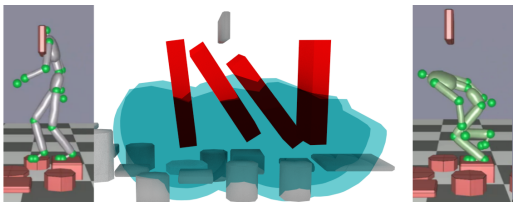


Figure 9: *Reachability planner.* ***Left****: The insertion of the top obstacle leads to invalidation of the deformation.* ***Middle:*** *The reachability RRT is used to compute a new trajectory for the root.* ***Right****: Using the proposed contact generator, a new motion, balanced and collision free, is generated.*

More importantly the search is biased to increase the likelihood of finding contacts compatible with the reference motion, while remaining complete.

7.1. The reachability condition

For a discrete frame, our objective is to find a new root configuration $\mathbf{q}^0$ satisfying three conditions: a) the resulting configuration is collision-free; b) the valid contacts remain valid; c) the invalidated contacts can be replaced. We formally define each condition on the configuration, from which we infer a reduced search space for our planner.

a) A necessary condition to collision avoidance. The valid root configuration is included in C^0_{free}:

$$C^0_{free} = \{\mathbf{q}^0 : W^0(\mathbf{q}^0) \cap O = \emptyset\} \tag{20}$$

where O is the set of obstacles in the environment, and W^0 is the bounding box of the character's head and torso.

b) A necessary condition to contact maintenance. To maintain existing contacts, the root configuration is included in C_{keep}:

$$C_{keep} = \{\mathbf{q}^0 : \forall \mathbf{p}^\mathbf{k}_\mathbf{c} \in \mathcal{C}, \quad \mathbf{p}^\mathbf{k}_\mathbf{c} \in W^k(\mathbf{q}^0)\} \tag{21}$$

where $\mathcal{C}$ is the set of contacts maintained.

c) A necessary condition for contact re-positioning. To be able to re-position a contact k, the root configuration is included in C^k_{rep}:

$$C^k_{rep} = \{\mathbf{q}^0 : W^k(\mathbf{q}^0) \cap O \neq \emptyset\} \tag{22}$$

Therefore, the search space for valid root positions verifies the reachability condition, given by $C_{reach} = C^0_{free} \cap C^k_{rep} \cap C_{keep}$. The reachability condition is illustrated in Fig. 9: the red rectangular shape encapsulates the torso and head of the character, while the blue shapes denote the reachability volumes of its legs.

7.2. Biased generation of a root path

To adapt a root trajectory, we thus implement a standard RRT planner, where the configuration generation is modified to only output paths contained in C_{reach}, without loss of generality. Testing the inclusion to C_{reach} is inexpensive, hence we simply generate candidate root configurations in the neighborhood of the original motion, and consider those that satisfy the reachability condition.

7.3. Full body motion generation

When a substitute root path is found by the planner, it is converted into a trajectory for the complete character. First we try to re-position the invalidated contacts as in Section 7. The other end-effector trajectories are computed to track those of the reference motion. To do so we use an inverse-kinematics scheme for each limb separately. This local adaptation is prone to failure (i.e. collision, joint limits, balance) but is cheap and keeps a large part of the style of the reference movement. In case of failure, the problematic contacts are re-positioned using the above method. The adaptation may arbitrarily loop between contact re-positioning and root re-planning but ultimately may fail, which marks the limits of our method.

Our method is thus able to not only re-position contacts upon deformation, but also to modify the root trajectory under dramatic environment updates, while preserving the original motion style.

8. Results

We demonstrate our approach in three different environments. Several examples in each context are demonstrated in the companion video, that is an important complement to the figures shown in the paper. All the examples reported in the video have been recorded in real-time: adaptation, re-positioning and re-planning are performed on-line while the user manipulates the environment geometry.

8.1. Car-ingress environment

This environment was used in [AAKC13]. We use it to show how our contribution extends the approach. In Fig. 10, the user changes the car geometry by elevating the roof of the car. The motion is adapted accordingly, the left hand contact position gets higher too, and the arm enters in collision with the door frame. This situation cannot be handled correctly by any previous adaptation method.

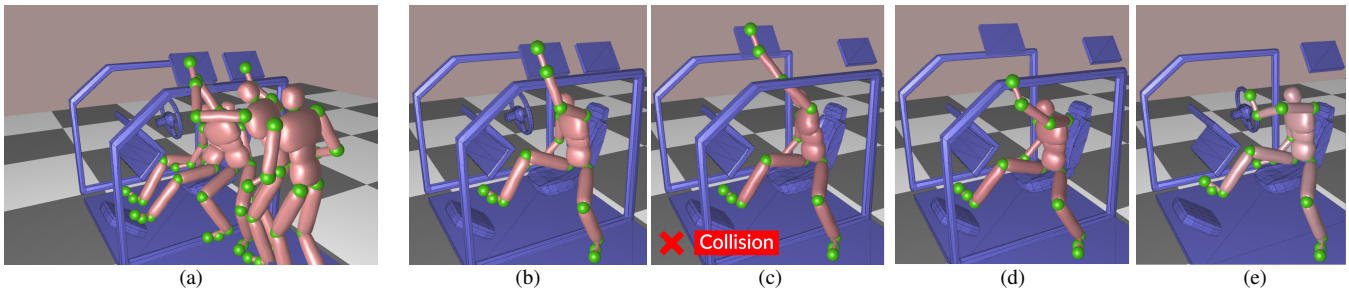
The collision is detected by our system, that automatically re-positions the hand contact on the door frame. The user makes the problem even more difficult by suppressing the door frame and roof geometries. Our solution successfully re-positions the left hand contact, finding that the turning wheel is now the best option.

8.2. Step stones

This environment demonstrates how our method handles locomotion tasks. In the real capture environment, several landmarks were placed on the floor at different heights, leading to a three-step ladder. The human actor stepped on the landmark before climbing the stepladder. The resulting motion capture is illustrated in Fig. 11-left.

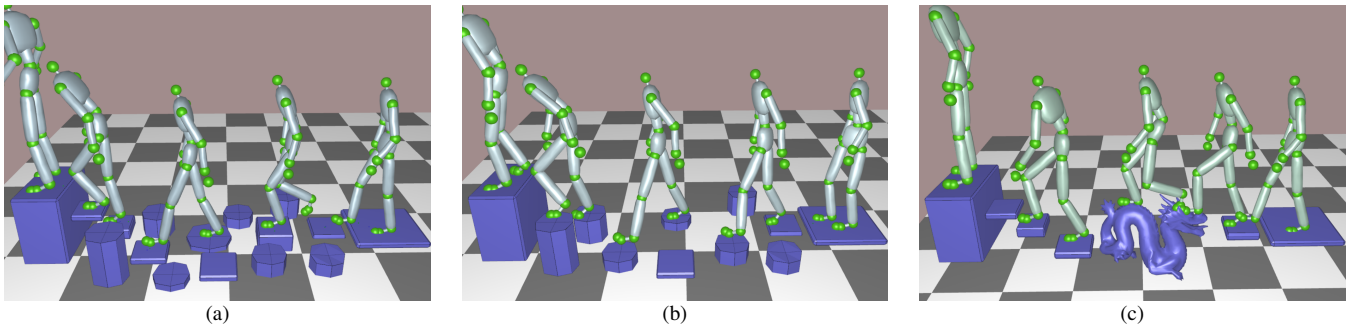
The relationship descriptors handle local deformations of the environment (positions and heights of the stones and ladder). Large deformations of the step stones bring the adaption to its limit (Fig. 1-left): a large translation of the green stone results in a unnatural motion. The proposed method detects this situation and re-positions the contact, putting the left foot on the green cylinder. As we show in the video, the transition between the adapted movement and the rest of the sequence is smooth and natural.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.



(a) (b) (c) (d) (e)

Figure 10: *Car-ingress movement.* ***(a)*** *Original captured motion (4 postures).* ***(b)*** *Original hand position with original car geometry.* ***(c)*** *For large environment changes, the local adaptation fails (here because of a collision).* ***(d)*** *Successful contact re-positioning.* ***(e)*** *The re-positioning enables the character to find a valid motion despite larger changes in the environment (cabriolet-like car).*



(a) (b) (c)

Figure 11: *Step-stone movement.* ***(a)*** *Original captured motion (5 postures) showing reference contact positions.* ***(b)*** *Adaption after several stones have been relocated. The second and fourth contacts have been re-positioned.* ***(c)*** *The Stanford dragon replaces several previous stones. New contacts are generated to step on this geometry.*

The method can handle very large deformations of the environment, as shown in Fig. 11, (middle and right images). Even when all the original step stones have been removed or the shape of the contacts are drastically changed.

8.3. Chair-table environment

The real capture environment is composed of a table and an armchair: the actor sits, interact with the table and gets up to walk away (Fig. 12-left). Complex interactions occur in this cluttered environment, while the character often change contacts to take support on the armrests or the table.

This example demonstrates that our method is robust to large changes in the geometry of the environment. The user sequentially suppresses the armrests and the table. The missing contacts are replaced by other objects of very different dimensions. The character uses the cylinder on its left hand when the left armrest is removed, or even re-position both contacts on the chair when needed. Our method is robust to these large changes and re-position hands correctly. The new adapted motions are plausible and continuous.

9. Discussion and Limitations

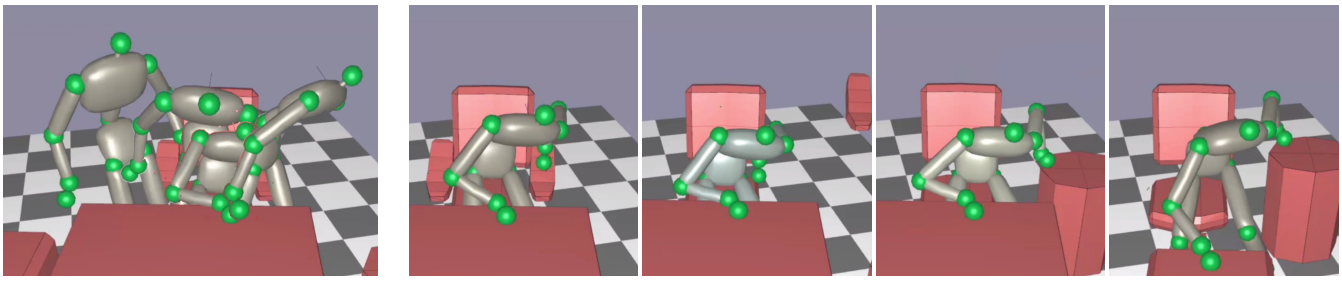
Motion plausibility. The quality of the motion we generate is a major aspect of our work. The motion is adapted progressively to the environment deformations, in a manner designed to preserve motion plausibility at best. Our work is based on the hypothesis that preserving the geometrical and physical interaction context of a reference motion is essential to generate plausible motion adaptations when contact re-positioning is required.

As a limitation to motion plausibility, we do not consider any semantic information in the contact re-positioning phase. In the car environment, our companion video shows that a large displacement of the turning wheel generates new contacts with a dashboard, but these contacts can have a different meaning.

Robustness. As shown in the Fig. 2, our method can lead to failure cases under extreme user manipulations of the environment geometry. There cannot be a guarantee that our method will find an acceptable motion adaptation, even if a solution may exist. However, we believe that our contact re-positioning strategy extends the possibilities of motion adaptation techniques. Indeed, few methods are able to automatically detect failure cases and find the appropriate contact replacements in the specific context we address.

For larger deformations, we recover new contacts using a sampling based planner (illustrated in the chair-table environment in the video). This insertion of purely-procedural elements can, in some case, lead to less plausible motions. This explains why the reachability RRT is used in the last step of our method.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.



(a) (b) (c) (d) (e)

Figure 12: *Chair-table environment.* ***(a)*** *Original captured motion (4 postures).* ***(b)*** *Initial posture for standing up. The next images show the re-positioning obtained after several deformations:* ***(c)*** *both armrests are removed, the character makes contact with the chair seat;* ***(d)*** *the character takes advantage of a new obstacle;* ***(e)*** *the table is lowered and the cylinder enlarged.*

9.1. Performances

Our solution is interactive, with up to one-second to adapt the motion in the most complex cases. Our demonstration scenarios are representative of typical usage in terms of number of contacts and environment complexity.

The performance of the method is however sensitive to the number of triangles in the scene (when checking for collisions and searching for new contacts). Before considering performance evaluation in future work, it would be relevant first to improve our implementation by designing a parallel version of the contact re-positioning part.

10. Conclusion

We present a method to interactively adapt motion capture animations when major changes occur in the environment geometry. Our method extends previous work, in particular [AAKC13], in two novel manners:

We propose objective criteria to determine when a adaption scheme produces unnatural motions (because of collision, joint limit violation, or loss of balance). When this failure happens, we propose a procedural scheme that allows to re-position the contacts involved in the motion. To do so, we introduce the force actuation profile, a heuristic that allows us to replace an invalid contact with an equivalent one regarding its contribution to the motion. Upon drastic changes, we ultimately rely on a biased sampling strategy, the reachability based planner, to adapt the root trajectory.

Building on the relationship descriptors method, we propose an optimization scheme that guarantees despite contact and root re-positioning, the motion remains continuous and smooth.

The ability to re-position contacts extends the amplitude of changes a user can make to the environment geometry. The current work immediately yields two possible applications. The first one is scene editing. An animator can adapt a geometry as he sees fit, without requiring to manually edit new contact positions when necessary. Furthermore, the performances of the method allow him to see the results of his edition immediately, thus allowing for an interactive editing loop. The second one is real time applications. In video games, a limited set of motion capture animations is used in various scenes. Those scenes are however often very similar due to the limitations of the methods of the industry. Contact re-positioning allows to introduce more variations in the scenes and is less restricting to the level designer creativity. For instance, the standing up from a chair motion that we demonstrate can be automatically played with a large variety of chairs, with or without armchairs for instance, thus allowing for a larger diversity. It would be interesting to experimentally evaluate the gain in animation time and creativity this can generate through a user-study.

A challenging issue to address for future work is the adaptation of a motion to sudden changes in the environment. This involves being able to find a continuous solution ahead of time, that anticipates the future state of the system. A promising direction is to combine our method with Model Predictive Control to achieve this, as proposed by [HRL15].

As a last remark, another direction for future research would be to isolate each contact interaction and consider the couple surface /limb independently from the rest of the body. Following the idea of motion graphs, we could then analyze long motion sequences to design "interaction graphs". This approach would allow us to decompose the motions and generate contact phases independently for each limb, thus addressing the combinatorial issue of contact creation in complex environments.

Acknowledgements

This research is supported by Euroc (project under FP7 Grant Agreement 608849), EU FP7/TOMSY, and Entracte (ANR grant agreement 13-CORD-002-01).

References

[AAKC13] AL-ASQHAR R. A., KOMURA T., CHOI M. G.: Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2013), SCA '13, ACM, pp. 45–53. URL: http://doi.acm.org/10.1145/2485895.2485905, doi:10.1145/2485895.2485905. 2, 3, 5, 8, 10

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications* (New York, NY, USA, 2001), SMA '01, ACM, pp. 249–266. URL: http://doi.acm.org/10.1145/376957.376986, doi:10.1145/376957.376986. 6

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

[AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM Trans. on Graphics 21*, 3 (2002), 483–490. 2

[BB04] BAERLOCHER P., BOULIC R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer 20*, 6 (2004), 402–417. 2

[CLS03] CHOI M. G., LEE J., SHIN S. Y.: Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Trans. on Graphics 22*, 2 (2003), 182–203. doi:10.1145/636886.636889. 2

[HET*14] HÄMÄLÄINEN P., ERIKSSON S., TANSKANEN E., KYRKI V., LEHTINEN J.: Online motion synthesis using sequential monte carlo. URL: http://doi.acm.org/10.1145/2601097.2601218, doi:10.1145/2601097.2601218. 2

[HKT10] HO E. S. L., KOMURA T., TAI C.-L.: Spatial relationship preserving character motion adaptation. *ACM Trans. Graph. 29*, 4 (July 2010), 33:1–33:8. URL: http://doi.acm.org/10.1145/1778765.1778770, doi:10.1145/1778765.1778770. 2

[HRL15] HÄMÄLÄINEN P., RAJAMÄKI J., LIU C. K.: Online control of simulated humanoids using particle belief propagation. In *Proc. SIGGRAPH '15* (New York, NY, USA, 2015), ACM. 2, 10

[Jak01] JAKOBSEN T.: Advanced character physics. *In Game Developers Conference Proceedings* (2001), 383–401. 11

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. In *ACM Trans. on Graphics* (New York, NY, USA, 2002), vol. 21, ACM. 2

[KHKL09] KIM M., HYUN K., KIM J., LEE J.: Synchronized multi-character motion editing. *ACM transactions on graphics (TOG) 28*, 3 (2009), 79. 2

[LCL06] LEE K. H., CHOI M. G., LEE J.: Motion patches: building blocks for virtual environments annotated with motion data. *ACM Trans. Graph. 25*, 3 (2006), 898–906. URL: http://doi.acm.org/10.1145/1141911.1141972, doi:10.1145/1141911.1141972. 2

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM Trans. Graph. 21*, 3 (July 2002), 491–500. URL: http://doi.acm.org/10.1145/566654.566607, doi:10.1145/566654.566607. 2

[LKJ99] LAVALLE S. M., KUFFNER J. J., JR.: Randomized kinodynamic planning, 1999. 7

[LLKP11] LEVINE S., LEE Y., KOLTUN V., POPOVIĆ Z.: Space-time planning with parameterized locomotion controllers. *ACM Transactions on Graphics (TOG) 30*, 3 (2011), 23. 2

[MC12] MIN J., CHAI J.: Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 153. 2

[MTP12] MORDATCH I., TODOROV E., POPOVIĆ Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. on Graphics 31*, 4 (2012). 2

[PBvdP15] PENG X. B., BERSETH G., VAN DE PANNE M.: Dynamic terrain traversal skills using reinforcement learning. *ACM Transactions on Graphics (to appear)* (2015). 2

[PLS03] PETTRÉ J., LAUMOND J.-P., SIMÉON T.: A 2-stages locomotion planner for digital actors. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), SCA '03, Eurographics Association. 2

[QEMR11] QIU Z., ESCANDE A., MICAELLI A., ROBERT T.: Human motions analysis and simulation based on a general criterion of stability. In *Int. Symposium on Digital Human Modeling* (2011). 6

[RBC98] ROSE C., BODENHEIMER B., COHEN M. F.: Verbs and adverbs: Multidimensional motion interpolation using radial basis functions. *IEEE Computer Graphics and Applications 18* (1998), 32–40. 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2004), ACM, pp. 175–184. 3, 7

[SH07] SAFONOVA A., HODGINS J. K.: Construction and optimal search of interpolated motion graphs. *ACM Trans. Graph. 26*, 3 (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276510. 2

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Transactions on Graphics (TOG) 20*, 2 (2001), 67–94. 2

[Tho91] THOMAS J. J. (Ed.):. *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991* (1991), ACM. 2

[TMP*15] TONNEAU S., MANSARD N., PARK C., MANOCHA D., MULTON F., PETTRÉ J.: A reachability-based planner for sequences of acyclic contacts in cluttered environments. In *Int. Symp. Robotics Research (ISRR), (Sestri Levante, Italy), September 2015* (2015). 7

[TPM14] TONNEAU S., PETTRÉ J., MULTON F.: Using task efficient contact configurations to animate creatures in arbitrary environments. *Computers & Graphics 45*, 0 (2014). 3, 5, 6

[vdPL95] VAN DE PANNE M., LAMOURET A.: Guided optimization for balanced locomotion, september 1995. URL: http://maverick.inria.fr/Publications/1995/VL95. 2

[WP95] WITKIN A., POPOVIC Z.: Motion warping. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 105–108. doi:10.1145/218380.218422. 2

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

[Yos84] YOSHIKAWA T.: Analysis and Control of Robot. Manipulators with Redundancy, 1984. 4

A1: Relative Vector Weights

Here we describe how we compute the weights in Eq. (1) that determine how each relative vector contributes to the joint position. For each descriptor points we first compute:

$$w'_{i,j} = \frac{\mathbf{n_i} \cdot \mathbf{r}_{i,j}}{\|\mathbf{r}_{i,j}\|}. \tag{23}$$

The weight fades out as the distance between $\mathbf{p_j}$ and $\mathbf{d_i}$ increases:

$$w''_{i,j} = w'_{i,j} f(\|\mathbf{r}_{i,j}\|), \tag{24}$$

$$f(d) = \begin{cases} 0 \ (r_2^j \leq d) \\ 1 - \frac{d - r_1^j}{r_2^j - r_1^j} \ (r_1^j < d < r_2^j) \\ 1 \ (d \leq r_1^j), \end{cases} \tag{25}$$

r_1^j is the distance to the closest descriptor point and r_2^j is set to $r_1^j + \frac{1}{4} \times bodyheight$. Finally, we normalize the weights:

$$w_{i,j} = \frac{w''_{i,j}}{\sum_i w''_{i,j}}. \tag{26}$$

A2: Fitting the Kinematic Skeleton to Target Joint Positions

Here we describe about the iterative inverse kinematics scheme based on [Jak01], where we fit the kinematic body structure to the

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

target joint positions obtained from Eq. (1). This scheme can be roughly broken down into the following four steps: (1) the computation of the affinity, which determines how strongly the joints must be pulled towards the target positions, (2) the force accumulation and integration step, where the effect of external forces such as pushing, pulling etc. are taken into account, and (3) the bone-length constraint step, where the joint positions are updated such that the distance between the adjacent joints are kept constant.

Affinity Calculation:

The affinity values are computed by summing the weights of the associated descriptors on the surface computed in Eq. (26) and normalizing them:

$$s_j = \frac{\sum_i^N w_{i,j}}{\sum_j^{N_j} \sum_i^N w_{i,j}} \tag{27}$$

where j is index of the joints and N_j is the number of joints.

Force Accumulation and Integration:

Instead of explicitly manipulating the joints, we control them by applying virtual forces to the particles that correspond to the joints. The forces are computed by multiplying an elastic constant to the difference between their current and target positions:

$$\mathbf{f}_j = k(\mathbf{p_j^{tar}} - \mathbf{p_j^{cur}}) + \mathbf{f^{ext}}. \tag{28}$$

where k is an elasticity constant that is set to 1, $\mathbf{p_j^{tar}}$ is the target position of the joint computed using the relationship descriptors by Eq. (1), $\mathbf{p_j^{cur}}$ is the current joint position, and $\mathbf{f^{ext}}$ is the external force that is added if an extra effect such as the wind blowing the body needs to be applied.

The joint target positions are then computed by Verlet integration:

$$\mathbf{p_j^{new}} = \mathbf{p_j^{cur}} + d(\mathbf{p_j^{cur}} - \mathbf{p_j^{prev}}) + \frac{1}{2}\mathbf{f_j}\frac{1}{N_s^2} \tag{29}$$

where $\mathbf{p_j^{prev}}$ is the position of the joint in the previous iteration and d is a coefficient that is added to reduce the wobbling effect whose value is set linear to the joint's affinity value ($d = 0.8$ when $s_j = 0$ and $d = 0.2$ when $s_j = 1$).

Constraints Step:

Using the updated particle positions $\mathbf{p_j^{new}}$, we compute the final positions of the joints that satisfy the bone-length constraint by iteratively updating the particle positions until the errors of all the constraints are below a certain threshold. To satisfy the bone-length constraints, the positions of each particle is updated by the following equation:

$$\Delta\mathbf{p_j} = \frac{s_k}{s_k + s_j}\frac{\mathbf{p_j} - \mathbf{p_k}}{\|\mathbf{p_j} - \mathbf{p_k}\|}(l^0 - \|\mathbf{p_j} - \mathbf{p_k}\|) \tag{30}$$

where $\mathbf{p_k}$ is a particle that is connected to joint j by a bone, and l^0 is the length of the bone. This will result in joints with large affinity to move less and small affinity to move more.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.