ORIGINAL ARTICLE

Motion constraint

Daniel Raunhardt · Ronan Boulic

Abstract In this paper, we propose a hybrid postural control approach taking advantage of data-driven and goal-oriented methods while overcoming their limitations. In particular, we take advantage of the latent space characterizing a given motion database. We introduce a motion constraint operating in the latent space to benefit from its much smaller dimension compared to the joint space. This allows its transparent integration into a Prioritized Inverse Kinematics framework. If its priority is high the constraint may restrict the solution to lie within the motion database space. We are more interested in the alternate case of an intermediate priority level that channels the postural control through a spatiotemporal pattern representative of the motion database while achieving a broader range of goals. We illustrate this concept with a sparse database of large range full-body reach motions.

Keywords Inverse kinematics · Motion editing · Posture control

1 Introduction

The applications of human-like character animation at interactive frame rates are manifold, ranging from computer

D. Raunhardt (✉) · R. Boulic
Ecole Polytechnique Fédérale de Lousanne, VRLAB Station 14, 1015 Lausanne, Switzerland
e-mail: daniel.raunhardt@epfl.ch

R. Boulic
e-mail: ronan.boulic@epfl.ch

entertainment to ergonomic studies or virtual prototyping. Human figures are difficult to animate such that they move in a coordinated and human-like fashion [7, 17]. We focus in the present paper on the problem of controlling a full-body goal directed motion (i.e. reach) where locomotion is not necessary.

Previous work has led to two different approaches for realistic character animation: data-driven and goal-oriented. Data-driven approaches create animations from motion captured data that contain the movement details from a human actor. The embedded natural flow of motion is their most attractive feature; on the other hand it can become tedious to generate movements that are outside the space of the captured movements. Goal-oriented techniques minimize the norm of postural variations and/or other physically-based cost functions while achieving a wide range of user-defined tasks. Their key advantage is their versatility at the cost of a frequent lack of naturalness when it comes to reproduce human activities.

In this paper we propose a hybrid approach aiming to combine the positive features of goal-oriented and data-driven approaches. Our goal is to benefit from the natural flow of movement provided by a motion database to channel the convergence of a Prioritized Inverse Kinematics (PIK) solver. This makes sense as the dimension of the user-defined goals are generally much smaller than the dimension of the joint space. Hence, even when exploited at a low priority, a motion database can influence each successive postural variation of the optimization convergence.

As already mentioned we use the action of *full-body reach* to illustrate the methodology throughout the paper; its generalization to other movement types is discussed at the end of the paper. We use a small number of reach motions where both feet remain fixed on the ground while the

Fig. 1 System architecture: the captured motions are used to build a low-dimensional, latent space using Principal Component Analysis (PCA). This pose PCA space is used to guide the PIK convergence by predicting its immediate future at low cost
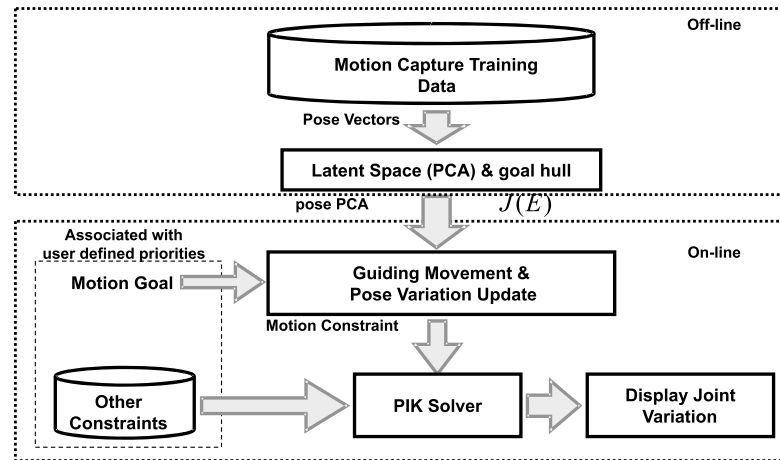
right hand reaches a specific goal in space (database provided by [19]). Such reach actions are highly coordinated movements that remain very similar independently of the goal location (front, side, back, low, high). They are the expression of a high-level motion synergy [5]. This allows to express them in a reduced latent space (e.g. Principal Component Analysis or other similar techniques).

We show in the paper how to specify a *motion constraint* (Fig. 1) in that space; its purpose is to generate a specific instance of a reach motion through a low-dimensional task. Its low dimension allows a transparent integration into the PIK framework. When assigned a high priority the motion constraint restricts the solution to lie within the motion database. Conversely, an intermediate priority level still allows it to guide the movement but a broader range of user-defined goals are feasible.

Finally, in accordance with our approach of interactive design, the proposed method still allows the user to alter any parameter of the PIK context, e.g. reach goal or priority etc., from one time step to the next.

Paper organization The reminder of this paper is organized as follows. Section 2 gives an overview of the literature. Section 3 presents a brief outline of the PIK solver used to solve constraints. Section 4 introduces the motion constraint. Finally, we present a summary of experimental results in Sect. 5 and discuss future work in Sect. 6.

2 Related work

Methods for modeling and generating human motions have been widely studied in biomechanics, robotics and computer graphics. Human motions are "under constrained" as there exist many ways to accomplish a task.

Data-driven approaches based on captured motion reproduce details and the style of human movement. Adapting

recorded motion to different characters or to new situations is often difficult [29]. Different approaches consist of blending between recorded motions [23], to edit motions [9] or producing new motions from the database [2, 16–18, 25]. However, most interpolation methods have difficulties handling constraints that do not interpolate or extrapolate from the data. Our proposed method overcomes these problems by combining learned motion data with an existing IK framework.

Goal-oriented approaches constrain the solution space to find natural-looking solutions in addition to enforce the specified user-defined goals [29]. IK is a popular technique to find a character pose that satisfies given constraints [3, 4, 10, 25, 29]. A number of techniques like energy consumption [12], mass displacement from a default pose [22] or distance to some reference pose [4] have been presented to restrict the IK solution space toward most "natural" postures. Rose et al. [23] explored the idea of using certain poses as a seed for IK to blend between motions using radial basis functions (RBF) changing their style. Kulpa et al. [17] proposed a motion representation independent of the character's morphology. With this representation, complex constraints can be rapidly enforced by applying a Cyclic Coordinate Descent algorithm. Grochow et al. [10] applied a nonlinear dimensionality reduction technique, called the Scaled Gaussian Process Latent Variable Model (SGPLVM) to human motion data. This learned, probabilistic model was combined with kinematic constraints to create new character postures.

Howe et al. [13] and Alexa et al. [1] used Principal Component Analysis (PCA) to reduce the dimensionality of motions. They represented the animations as a set of Principal Components. Glardon et al. [8] used PCA predictions to tune step constraints for walking, running and jumping motions. Following this idea, Safanova et al. [24] proposed a hybrid motion synthesis framework that created new motions by optimizing a set of constraints in the low-dimensional PCA

space. They used IK simply in the limbs of the virtual mannequin to prevent undesirable artifacts such as foot sliding. Yamane et al. [29] used a Rapidly-Exploring Random Tree (RRT) planner to perform path planning for an object being manipulated. They used IK to position the character and to satisfy simultaneously external constraints. They biased IK toward poses generated from pre-recorded motion captured data. Finally, they applied a path smoothing and computed velocity profiles. Aydin et al. [3] combined a database with forward kinematics and IK to determine appropriate poses for the hand to grasp an object. However, their method required a dense database and works off-line. Carvalho et al. [6] combined a PCA motion model with a PIK framework to solve external constraints in the low-dimensional PCA space. Their work strongly focused on traditional per-frame constraint-based motion editing. As we want to build a sparse latent space with different styles of motions and to change interactively the goal, we cannot use such a simple motion PCA space.

Park et al. [21] identified basic spatiotemporal structures of human motion and method for generalization. The identified structures of the sample motions could be applied to similar motion variants by concatenating joint-angle–time trajectories.

In this paper we present an algorithm for integrating motion synergies into an existing PIK framework. Its key characteristic is to coordinate the joint contribution [5] and to synchronize them on the motion flow pattern of recorded movements using a sparse database. Our work concentrates on reaching motions where some of the constraints (e.g. feet and wrists) are known in advance [3, 16, 23, 29]. We demonstrate, through several examples, that our algorithm is fast, robust and that the goal can interactively be changed.

3 Prioritized inverse kinematics

At the core of our system lies a PIK solver which allows us to find a configuration of an articulated structure (i.e., its joint state, Θ) given a task vector x constructed from a set of user-specified Cartesian constraints while joint limit avoidance is guaranteed [4]. The PIK ensures that a constraint associated with a high priority is achieved as much as possible. Those of lower priority are optimized only on the reduced solution space and do not interfere with higher priority constraints. After all constraints have been solved, the remaining degrees of freedom (DOF) can be exploited by a joint-space optimization function, like attracting toward a predefined rest posture.

4 Motion constraint

4.1 Overview

The main idea is to create a low-dimensional motion constraint from a sparse set of captured motions. We take advantage of the embedded spatiotemporal information available from the database to determine a posture variation.

Latent space Each motion captured posture is represented by a vector Θ_i where i is an index over the captured postures. We apply PCA on the vectors Θ_i to define a low-dimensional representation of the original data. This low-dimensional space of the Θ_i is called *latent space*. Every posture Θ_i has a corresponding representation in the latent space. Postures that belong to the same motion are chronologically connected in the latent space.

Latent space and trajectory deformation Given the user-defined goal position P_G for the controlled effector (i.e. the right wrist for the reach example), we deform the captured motions in our latent space. The deformed motion achieves the user-defined goal P_G while maintaining as much as possible the style and temporal grounding of the original motions.

Motion constraint We integrate the deformed motion transparently into our PIK framework to determine a posture variation while retaining the spatiotemporal information from the captured motions. This type of spatiotemporal constraint is called *motion constraint*. A motion constraint can be achieved at any priority level and it can be combined with any other Cartesian constraint.

4.2 Latent space

We define the 3D motion captured posture of a character with a vector Θ that consists of the global position and orientation of the root and all of the joint angles of the articulated structure:

$$\Theta = [\theta_1, P_1, \ldots, \theta_n], \tag{1}$$

where P_1 and θ_1 represent the normalized 3D global position and 3D orientation of the root node and θ_j is the local transformation of the jth joint expressed using the exponential map formulation [11]. By applying PCA, any posture vector Θ can be represented as the linear combination of:

$$\Theta \approx \overline{\Theta} + \sum_{i=1}^{m} \alpha_i E_i, \tag{2}$$

where $\overline{\Theta}$ is the mean posture, E_i are the Principal Components computed by the PCA and $\alpha = (\alpha_1, \ldots, \alpha_m)$ are the
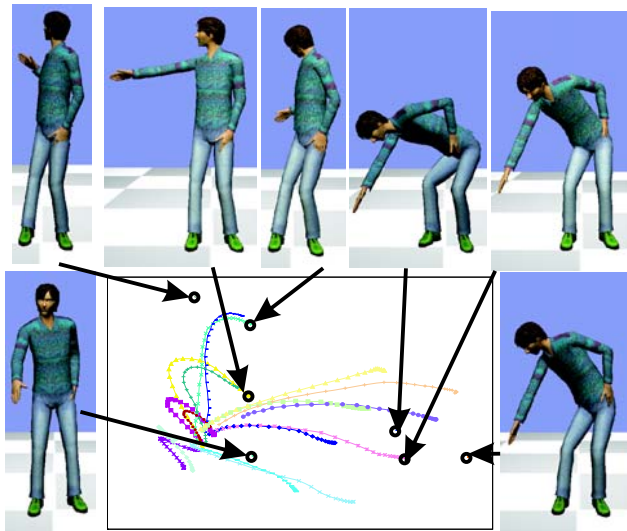
Fig. 2 PCA latent space with 16 learned reach motions as well as some extrapolated postures. Points indicate training poses included in the training set. Points that are connected belong to the same motion sequence. We present here only the first two Principal Coefficients

Principal Coefficients that characterize the motion. m represents the number of Principal Components that are required to reconstruct a desired percentage of the database [14]. All postures of a motion are connected chronologically inside this latent space.

We use a sparse motion database of 16 different right wrist reach motions provided by [19]. In Fig. 2 we show the PCA latent space constructed from a few reach sequences. We normalize the motion duration such that every motion has the same importance for building the latent space. The Principal Components as well as the Principal Coefficients are estimated off-line as these parameters stay constant.

4.3 Latent space and trajectory deformation

In this section, we describe how to compute a motion in the latent space that achieves the user-defined reach motion goal P_G.

In our examples, we focus on constraining the right wrist to a specific location in space P_G. The user-specified reach motion constraint has to reproduce the type of motions in the latent space. To use the same motion database with different body heights, we use a scaling factor that maps the current body height to the body height of the captured postures. This simple scaling factor only works with different body heights that have the same body proportions (see Fig. 3d and Fig. 3e).

To compute a weighted combination of captured motions that achieve P_G in the latent space, we first express the user-defined goal P_G as a combination of its nearest neighbors in

Cartesian space using barycentric coordinates [28]:

$$P_G = \sum_{i=0}^{3} b_i P_i, \tag{3}$$

where b_i are the barycentric coordinates and P_i are vertices that build a tetrahedron enclosing P_G. The vertices P_{1-3} correspond to captured, final positions while the fourth vertex P_0 is set to the initial, rest position of the controlled effector. The final positions build a goal hull that covers the whole space of captured motion for the controlled effector. In our examples, the *goal hull* is given by the final position of the right wrist of the captured motions (Fig. 3a). Our choice of P_0 allows us to build a set of tetrahedrons that are never intersecting. Thus, we can change the goal interactively while the goal P_G can be expressed continuously using the barycentric coordinates b_{0-3}. We allow negative values of b_0 such that we can guarantee that the barycentric coordinates exactly achieve the goal P_G even if the goal is outside the goal hull. We obtain a deformed trajectory of the controlled effector by applying the barycentric coordinates to the motions associated with the tetrahedron enclosing P_G.

The barycentric coordinates b_{0-3} that define the goal P_G in Cartesian space are used as a first estimation of the final state in the latent space. The final state in the latent space given by the barycentric coordinates b_{0-3} can be represented as:

$$\widetilde{\alpha_G} = \sum_{i=0}^{3} b_i \alpha_{P_i}, \tag{4}$$

where α_{P_i} are the Principal Coefficients of the vertex P_i. However, there is no guarantee that the corresponding goal position $\widetilde{P_G}$ of $\widetilde{\alpha_G}$ exactly matches P_G. To refine the barycentric coordinates, we relate their variation Δb to a variation Δx of the goal P_G:

$$\Delta b = J(b)^{+\lambda} \Delta x \tag{5}$$

with the damped pseudo-inverse of the Jacobian $J(b) = [\delta x/\delta b]$. Owing to the chain rule, the Jacobian $J(b)$ is defined as:

$$\left[\frac{\delta x}{\delta b} \right] = \left[\frac{\delta x}{\delta \theta} \right] \left[\frac{\delta \theta}{\delta \alpha} \right] \left[\frac{\delta \alpha}{\delta b} \right]. \tag{6}$$

Besides, the $[\delta\theta/\delta\alpha]$ is the Jacobian $J(E)$ of the latent space built by the Principal Components E_{1-m}. Thus, $J(E)$ is constant for a given motion database. The $[\delta\alpha/\delta b]$ relates a change of the barycentric coordinates to a change in the latent space. As long as the vertices P_{1-3} remain the same (i.e. the user-specified goal stays within the same tetrahedron), $[\delta\alpha/\delta b]$ is constant and only the Jacobian $[\delta x/\delta\theta]$ has to be updated. To refine the barycentric coordinates b_{0-3}, we
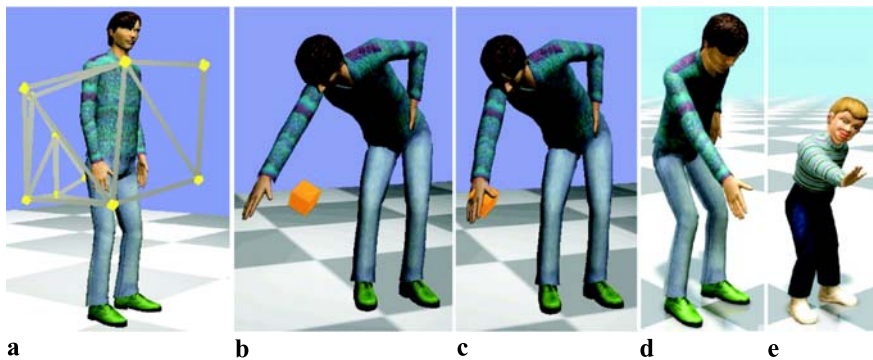
Fig. 3 (**a**) Goal hull of right wrist built by end-positions of the captured motions indicated by a *yellow cube*. (**b**) Estimating the final reaching posture using b_{0-3} of a goal on the right indicated by a *yellow cube*. (**c**) Estimated final posture of the refined barycentric coordinates using w_{0-3} of a goal on the right indicated by a *yellow cube*. (**d**), (**e**) Motion constraint applied to two different body heights reaching the same goal in Cartesian space

Fig. 4 Overview for computing the barycentric coordinates and how to extend the solution space. Given the barycentric coordinates b_{0-3} we compute the barycentric coordinates w_{0-3} to find a deformed motion in the latent space that achieves the user-defined goal P_G
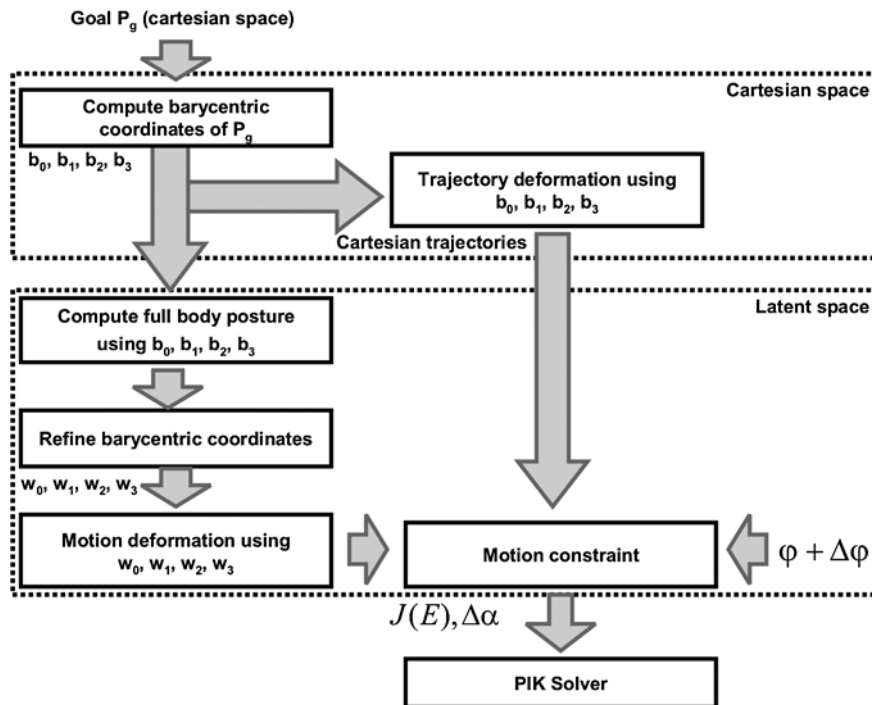
iteratively use (5) converging to the refined barycentric coordinates w_{0-3} of the final state in the latent space. This small convergence loop is very efficient as $J(b)$ is only a 3×3 matrix.

We still cannot guarantee that the refined barycentric coordinates w_{0-3} exactly achieve the goal P_G (Fig. 3b and Fig. 3c). But the barycentric coordinates w_{0-3} define a motion deformation in the latent space such that the user-specified goal is achieved as much as possible while retaining the spatiotemporal coherence of the captured motions. Figure 4 presents an overview for computing the barycentric coordinates. Thus, we have two different sets of barycentric coordinates: b_{0-3} and w_{0-3}. The b_{0-3} deform the Cartesian trajectory of the controlled effector while w_{0-3} are used

for motion deformation in the latent space to take advantage of its low-dimensional spatiotemporal coherence. If a user interactively changes the goal P_G, we have to update the barycentric coordinates b_{0-3} of the Cartesian trajectory as well as w_{0-3} in the latent space.

4.4 Motion constraint

In this section we describe how to integrate the methods discussed in the previous sections into a PIK framework that works on the joints variations.

A motion constraint has to handle two key aspects: reaching P_G as well as preserving the temporal pattern of the motion synergy. Given a goal P_G, an objective is to guide

the PIK convergence by using the latent space reproducing the style of the captured motions while reaching the user-specified goal P_G. We have the general IK formulation determine a posture variation that satisfies the specified constraints:

$$\Delta\Theta = J(\Theta)^{+\lambda}\Delta x, \tag{7}$$

where $J(\Theta)^{+\lambda}$ is the damped pseudo-inverse of the Jacobian matrix $J(\Theta)$. Our motion constraints work in the latent space and not in the Cartesian constraints task space. Due to a chain rule similar to (6) we obtain the following equation:

$$\Delta\Theta = J(E)^{+\lambda}\Delta\alpha, \tag{8}$$

where $J(E)^{+\lambda}$ is the damped pseudo-inverse of the Principal Components (latent space). Equation (8) relates a variation $\Delta\alpha$ in the latent space to a joint variation $\Delta\Theta$; the motion constraint can be used as any other Cartesian constraint. The user may also specify the set of joints that are influenced by the motion constraint to achieve the solution. It is important to note that the Principal Components are estimated off-line and that they remain constant for a given latent space.

4.5 Synchronization

For traditional Cartesian constraints, the task to achieve is to minimize the distance between the end-effector and the goal. For a motion constraint such a simple definition is not appropriate as we want the PIK framework to follow the deformed motion in the latent space. This leads us to another aspect of the motion constraint: the management of the temporal dimension. We use two parameters to control the temporal dimension of a motion constraint. The first parameter is the *current phase* φ and the second is the *phase increment* $\Delta\varphi$. As the deformed motion in the latent space is built by motions with different durations, we normalize φ between $[0, 1]$ to provide an easy mapping from the motion constraint to the real durations of the captured motions. By construction, the motion constraint has accomplished its task at $\varphi = 1$. An estimation of the real motion duration is given by the barycentric coordinates w_{1-3} and the duration of the corresponding captured motions. This information can be useful for creating animations. The user-defined phase increment $\Delta\varphi$ defines the step size that controls the speed. It defines how fast we want to follow the deformed motion. The motion constraint evaluates the Principal Coefficients α_Θ of the current joint configuration Θ, the Principal Coefficients of the deformed motion α_{DM} at phase $\varphi + \Delta\varphi$ and computes the difference $\Delta\alpha_G = \alpha_{DM}(\varphi + \Delta\varphi) - \alpha_\Theta$. The damped pseudo-inverse then computes the optimal joint variation solution to achieve the goal $\Delta\alpha_G$. After having computed the joint variation, we update the phase by $\varphi = \varphi + \Delta\varphi$. Thus,

the motion constraint continuously follows the deformed motion guiding the PIK optimization process. The previously made considerations about the management of the temporal dimension have to be slightly modified when constraints other than the motion constraint are present. More specifically, the motion constraint can lie on any priority level. Higher level priority constraints can prevent the motion constraint to accomplish its desired goal. Thus, the update of the phase by $\varphi = \varphi + \Delta\varphi$ is no longer necessarily valid and we cannot ensure that the motion constraint exactly follows the deformed motion in the latent space. To overcome this problem, we have to synchronize the motion constraint with the current joint configuration. First we compute the Principal Coefficients α_Θ of the current joint configuration Θ. To synchronize φ, we simply set φ to the phase of the closest posture of the deformed motion in the latent space. To reduce the search space, we compare α_Θ only with postures in the latent space that have been achieved at a similar phase as our current φ. We enforce that the synchronized φ cannot be smaller than it was before. Depending on the synchronization interval, the same φ can be evaluated during several convergence iterations of the PIK framework. If the phase is synchronized or the user-defined goal changes, we also re-compute the barycentric coordinates.

4.6 Extending the solution space

The balance of the virtual mannequin is automatically ensured by the motion constraint. Due to other higher priority constraints, the motion constraint may not be exactly achieved and therefore it cannot guarantee the balance. Another problem is that the 3D position $\widetilde{P_G}$ of the estimated posture is not necessarily the same as the user-specified position P_G. This mainly depends on the density of the database. To overcome these problems, we propose to combine the motion constraint with a Cartesian position constraint for the constrained effector as well as with a time dependent center of mass (COM) constraint given the mass distribution of the articulated structure. The deformed trajectories of the constrained effector and the time dependent COM constraint are given by the barycentric coordinates b_{0-3}. They ensure that the deformed effector trajectory exactly reaches the user-specified position while balance is maintained. We use the same phase parameters φ and $\Delta\varphi$ as for the motion constraint to control the desired position of the constrained effector and the COM constraint over time. We set the priorities of these effectors as follows: time dependent COM constraint > time dependent Cartesian position constraint > motion constraint to avoid discontinuities and to guarantee that the user-specified goal only is exactly achieved (if possible) while strictly keeping balance of the virtual mannequin. Thus, we are able to precisely manage over time the controlled effectors due to the deformed trajectories as well
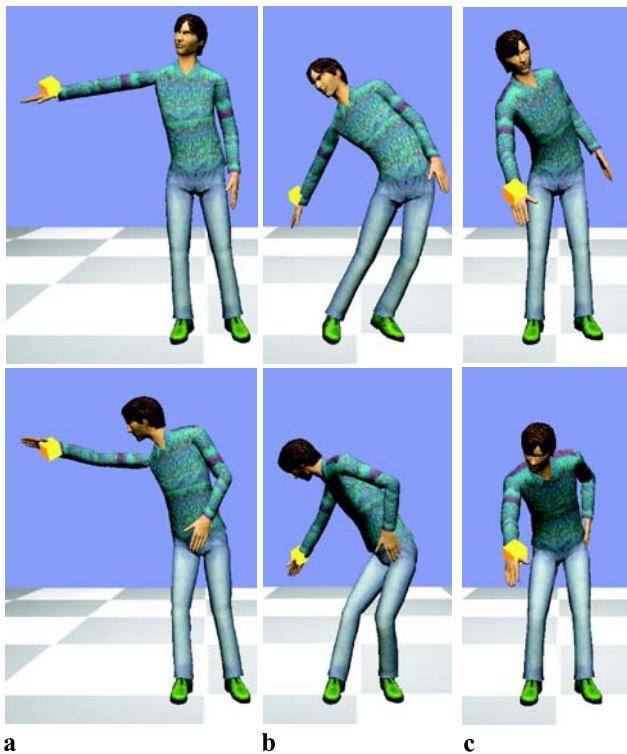
Fig. 5 Results for three different reaching motions for a goal outside the goal hull. The goal is indicated by a *yellow cube*. *Upper row*: Prioritized Inverse Kinematics. *Lower row*: motion constraint. (**a**) Goal on the side. (**b**) Goal in the back. (**c**) Goal in the front

as take advantage of the full-body spatiotemporal coherence using the motion constraint (Fig. 5 and Fig. 6).

4.7 Summary

To summarize, our motion constraint is composed of a deformed motion in the latent space preserving the temporal pattern of the motion synergy. To deform the captured motions, we use barycentric coordinates. Through phase parameters, the user can control the speed of the deformed motions. As we cannot guarantee that the deformed motion exactly reaches the user-specified goal, we propose to add an additional Cartesian constraint to control the effector trajectory that uses the same phase parameters as for the motion constraint. By combining the motion constraint with an additional Cartesian constraint and a time dependent COM constraint, we extend the solution space while retaining the spatiotemporal coherence of the captured motions.

5 Results

5.1 Final posture comparison

This section presents the results for a variety of difficult reaching tasks. The motion constraint database used for all

Table 1 Basic constraint set used during our experiments

Constraint type	Controlled body part	Priority rank	Standard PIK	PIK with motion constraint
Center of mass	Root	1	x	–
Position	Feet	2	x	x
Position	Right wrist	3	x	x
Motion constraint	All joints	4	–	x

examples is described in Sect. 4.2 (16 reach motions, each containing 25 postures). We use 95% of the information retrieved from the PCA. Thus, a posture is represented by only 6 Principal Coefficients. Table 1 defines the different basic types of constraints that are used in the following experiments.

The center of mass constraint is used to ensure the balance of the virtual mannequin given the mass distribution of the articulated structure. The balance is automatically ensured through the motion constraint. Thus, we do not use the center of mass constraint together with the motion constraint. For example motions with the time dependent COM constraint, see the accompanying video. We compared two approaches: standard PIK and PIK with the proposed motion constraint.

We executed three experiments with difficult reach goals for the right wrist: once the goal is on the side (Fig. 5a), in the back (Fig. 5b), and in the front (Fig. 5c). Every goal is outside the goal hull built by the captured motions and the distance between the desired goal and the closest corresponding position in the database varies between 0.25 m (front) and 0.4 m (back). For the goal in the front and on the right side, the center of mass is placed under the right foot. For the goal in the back, the center of mass constraint is placed under the left foot. Comparing the results in Fig. 5, the motion constraint helps to converge toward more realistic postures.

In general, an important aspect with IK is the number of convergence iterations required to reach the goal. As we can control the speed of the motion constraint through its task parameter $\Delta\varphi$, we implicitly control the number of iterations required to reach the goal. For the normal PIK we do not have such control.

In the accompanying video, we compare virtual mannequins with different body heights that have to reach the same goal in Cartesian space. We also present two motions combining the motion constraint including the time dependent COM constraint with other constraints (i.e. holding a bottle vertical while reaching a goal).
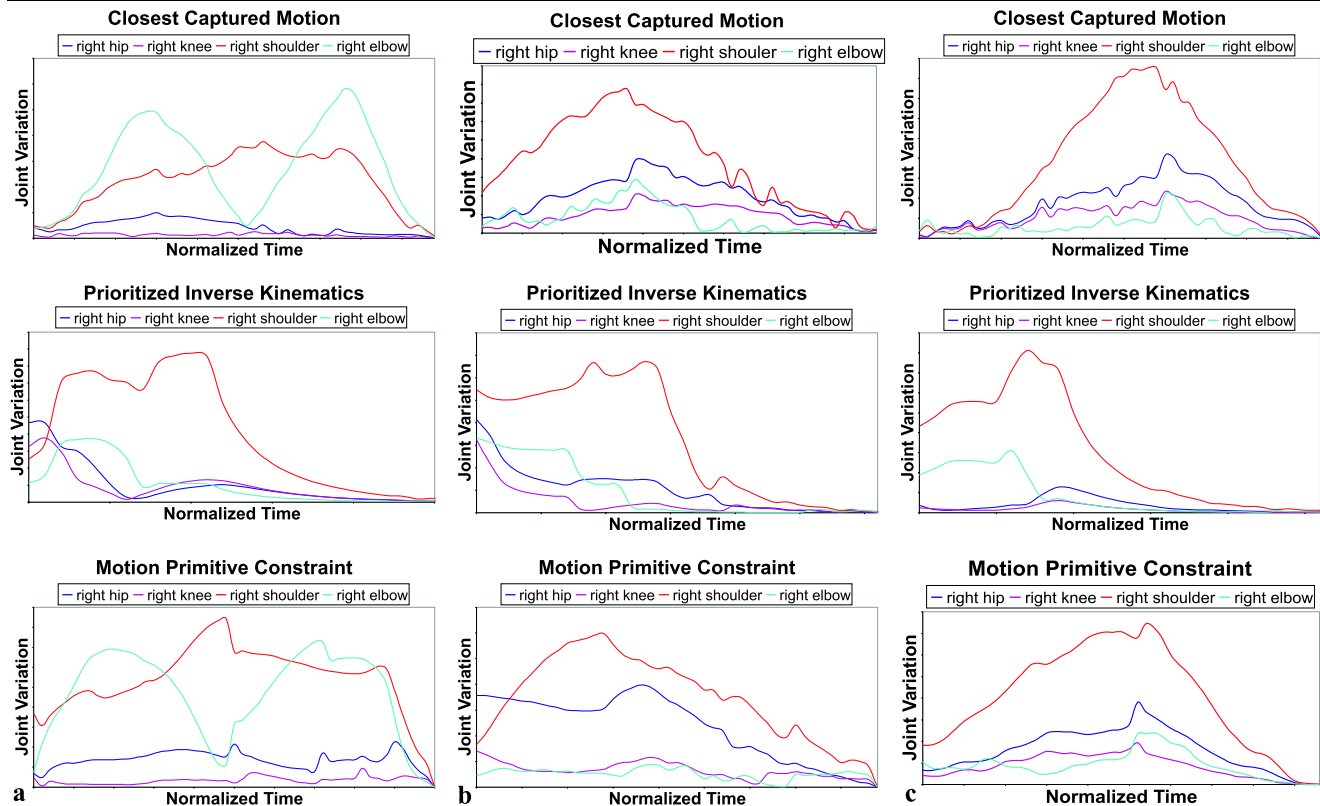
Fig. 6 Joint variations for the three different reaching goals. *Upper row*: closest captured motion. *Middle row*: Prioritized Inverse Kinematics. *Lower row*: motion constraint. (**a**) Goal on the side. (**b**) Goal in the back. (**c**) Goal in the front

5.2 Temporal pattern comparison

When achieving a reach goal, the joint contributions follow a characteristic temporal pattern [15, 26, 27]. For example there exists a high correlation behavior in the joint-angle space between the elbow and the shoulder. The standard PIK framework has no information about this correlation. It only finds a minimum norm in the joint configuration space that brings the wrist nearer to the solution. We have measured the joint variations (amplitude of the exponential map) of the right hip, right knee, right shoulder and right elbow. Figure 6 presents them for the closest captured motion, the standard PIK solution and our approach including motion constraint. The joint variations produced with the motion constraint capture many features of the closest captured motion. In general the elbow-shoulder correlations are well modeled by our motion constraint. However, there can be some discrepancies. The motion constraint usually starts with a larger joint variation than the captured motions. The reason for this is that our deformed motion is a combination of different captured motions whose latent space rest poses slightly differ from one another (Fig. 2). Thus, the motion constraint has to bring the initial posture to the initial posture of the deformed motion. The other key difference in the joint variations is due to our reaching locations outside the goal hull.

But as seen in Fig. 6 and in the accompanying video, our motion constraint succeeds in channeling the joint variations to reflect the original reach flow of movement.

5.3 Performance

The motion constraint consists in off-line and on-line computations. The Principal Coefficients of the postures, the Principal Components of the latent space as well as the goal hull and a set of non-intersecting tetrahedrons are computed off-line. Given the user-specified goal P_G, we have to find the barycentric coordinates b_{0-3} for the effector trajectory deformation and the refined barycentric coordinates w_{0-3} for the motion deformation. The computation of the barycentric coordinates for the effector trajectory deformation can be done in constant time using a spatial hashing between the goal P_G and the tetrahedrons.

Once the barycentric coordinates b_{0-3} of the deformed trajectory have been computed, we have to refine them. This requires a small convergence loop using (5) that relates a change of the barycentric coordinates to a variation of the goal P_G. We only have to re-compute the barycentric coordinates b_{0-3} and w_{0-3} when the user-specified goal changes. To synchronize the phase, we only have to compare the current pose with poses of motions corresponding to the

Table 2 Computational time for different update steps of the motion constraint measured on a Pentium Dual Core 3.2 GHz with 2 GB RAM

Update step	Computational time [μs]
Synchronize time	250
Barycentric coordinates effector trajectory deformation (computing b_{0-3})	14
Refining barycentric coordinates for motion deformation (computing w_{0-3})	457

barycentric coordinates w_{1-3}. To further reduce the number of poses, we only compare poses of the corresponding motions which are in a similar phase. If we are not interested in the joint variations over time, the motion constraint can be used to let the PIK optimization converge within a few iterations by setting $\Delta\varphi = 1 - \varphi$. Table 2 presents the computational time required for the different stages of our motion constraint.

6 Discussion

We have introduced a new type of constraint for IK that relies on a sparse motion database. Using barycentric coordinates, the algorithm deforms a motion in the latent space that achieves the user-specified final goal. Moreover, its speed to convergence can be controlled through a phase increment parameter by the user. The work required to generate a final posture is: (1) set the goal position for an effector that corresponds to the captured motions; (2) set the Cartesian constraints (e.g., feet on the ground). Our system then computes a deformed motion in the latent space that should achieve the desired goal. For extending the solution space, we propose to use an additional Cartesian constraint using a deformed effector trajectory and a time dependent COM constraint.

Due to our transparent integration of the motion constraint into a PIK solver, even multiple simultaneous motion constraints can be supported as the strict priority levels prevent them to interfere with one another. An example would be a reaching motion that includes a step. One motion constraint could control the upper body while another motion constraint is responsible for controlling the legs.

Our method implicitly assumes that there is a strong relation between the captured motions and the user-specified goal to be achieved by the motion constraint. To increase the accuracy of our motion constraint, a slightly larger set of captured motions can be used. We have shown that using a very sparse database, our motion constraint reaches goals far from the goal hull while retaining the spatiotemporal characteristics of the captured motions over time.

Future work We plan to extend our approach to higher dimensional motion synergies such as the generalized step (forward/backward, up/down, sideward, turn) and a generalized reach with an additional orientation constraint.

References

1. Alexa, M., Mueller, W.: Representing animations by principal components. Comput. Graph. Forum **19**, 3 (2000)
2. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. CM Trans. Graph. **22**(3), 402–408 (2003)
3. Aydin, A., Nakajima, M.: Database guided computer animation of human grasping using forward and inverse kinematics. Comput. Graph. **23**, 145–154 (1999)
4. Baerlocher, P., Boulic, R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. Vis. Comput. **20**(6), 402–417 (2004)
5. Berthoz, A.: The Brain's Sense of Movement. Perspectives in Cognitive Neuroscience. Harvard University Press, Cambridge (2002)
6. Carvalho, S.R., Boulic, R., Thalmann, D.: Interactive low-dimensional human motion synthesis by combining motion models and PIK. Comput. Animat. Virtual Worlds **18**(4–5), 493–503 (2007)
7. Egges, A., Molet, T., Magnenat-Thalmann, N.: Personalised real-time idle motion synthesis. In: Pacific Graphics, Seoul, Korea, pp. 121–130 (2004)
8. Glardon, P., Boulic, R., Thalmann, D.: Robust on-line adaptive footplant detection and enforcement for locomotion. Vis. Comput. **22**(3), 194–209 (2006)
9. Gleicher, M.: Retargetting motion to new characters. In: SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 33–42. ACM, New York (1998)
10. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. In: SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, pp. 522–531. ACM, New York (2004)
11. Grassia, F.S.: Practical parameterization of rotations using the exponential map. J. Graph. Tools **3**(3), 29–48 (1998)
12. Grassia, F.S.: Believable automatically synthesized motion by knowledge-enhanced motion transformation, PhD Thesis, Pittsburgh, PA, USA (2003)
13. Howe, N.R., Levention, M.E., Freeman, W.T.: Bayesian reconstruction of 3d human motion from single-camera vide. In: Advances in Neural Information Processing Systems 12, pp. 820–826. MIT Press, Cambridge (2000)
14. Jolliffe, I.T.: Principal Component Analysis. Springer, Berlin (1986)
15. Korein, J.U.: A Geometric Investigation of Reach. MIT Press, Cambridge (1985)
16. Kovar, L., Gleicher, M.: Automated extraction and parameterization of motions in large data sets. ACM Trans. Graph. **23**(3), 559–568 (2004)
17. Kulpa, R., Multon, F., Arnaldi, B.: Morphology-independent representation of motion for interactive human-like animation. Comput. Graph. Forum **24**, 343–352 (2005)
18. Lee, J., Chai, J., Reitsma, P., Hodgins, J.K., Pollard, N.: Interactive control of avatars animated with human motion data. ACM Trans. Graph. **21**(3), 491–500 (2002)

19. Mukai, T., Kuriyama, S.: Geostatistical motion interpolation. ACM Trans. Graph. **24**(3), 1062–1070 (2005)

20. Nakamura, Y., Hanafusa, H., Yoshikawa, T.: Inverse kinematic solutions with singularity robustness for robot manipulator control. J. Dyn. Syst. Meas. Control **108**, 163–171 (1986)

21. Park, W., Chaffin, D.B., Martin, B.J.: Toward memory-based human motion simulation: development and validation of a motion modification algorithm. IEEE Trans. Syst. Man Cybern., Part A **34**(3), 376–386 (2004)

22. Popovic, Z., Witkin, A.: Physically based motion transformation. In: SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 11–20. ACM, New York (1999)

23. Rose, C.F., III, Sloan, P.-P.J., Cohen, M.F.: Artist-directed inverse-kinematics using radial basis function interpolation. Comput. Graph. Forum **20**, 3 (2001)

24. Safanova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Trans. Graph. **23**(3), 514–521 (2004)

25. Shin, H.J., Lee, J.: Motion synthesis and editing in low-dimensional spaces, Research articles. Comput. Animat. Virtual Worlds **17**(3–4), 219–227 (2006)

26. Wang, X.: Three-dimensional kinematic analysis of influence of hand orientation and joint limits on the control of arm posture and movement. Biol. Cybern. **80**(6), 449–463 (1999)

27. Wang, X.: Behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation. J. Biomech. **32**(5), 453–460 (1999)

28. Weisstein, E.W.: Barycentric Coordinates, Mathworld—A Wolfram Web Resource

29. Yamane, K., Kuffner, J.J., Hodgins, J.K.: Synthesizing animations of human manipulation tasks. ACM Trans. Graph. **23**(3), 532–539 (2004)

Daniel Raunhardt received the M.Sc. in Computer Science from the Swiss Federal Institute of Technology, Zurich. Currently he is a Ph.D. student at the Swiss Federal Institute of Technology in Lausanne, where he focuses on handling high-dimensional motion synergies and inverse kinematics.

Ronan Boulic is a Senior Researcher, Lecturer and Ph.D. Advisor at the Ecole Polytechnique Fédérale de Lausanne (EPFL). He is working in the Virtual Reality Lab and his research interests include 3D interactions, motion capture, modeling, and synthesis for virtual humans and robots. He received the Ph.D. degree in Computer Science in 1986 from the University of Rennes, France, at the INRIA-IRISA research institute.