Motion Editing with Prioritized Constraints

Ronan Boulic, Benoît le Callennec, Martin Herren, Herbert Bay

VRlab, I&C

EPFL Lausanne CH-1015 Switzerland

Telephone: +41.21.69.352.46, Fax: +41.21.69.353.28

E-mail: ronan.boulic@epfl.ch, benoit.lecallennec@epfl.ch, martin.herren@epfl.ch, herbert.bay@epfl.ch

Abstract

We propose an intuitive motion editing technique allowing the end-user to transform an original motion by applying position constraints on freely selected locations of the character body. The major innovation comes from the possibility to assign a *priority level* to each constraint. The resulting scale of user-defined priority levels allows to handle multiple asynchronously overlapping constraints. As a consequence the end user can enforce a larger range of natural behaviors where conflicting constraints compete to control a common set of joints. By default the joint angles of the original motion are preserved as the lowest priority constraint. However, in case a Cartesian constraint from the original motion is essential, it is straightforward to define a high priority constraint that will retain it before enforcing other lower priority constraints. Additional features are proposed to provide a more productive motion editing process like defining the constraints relative to a mobile frame and precisely specifying which joints are recruited by each constrain. The current limitation remains its computing cost which will be improved in future work.

Keywords

Motion Representation, Motion Synthesis, Motion Editing, Prioritized Inverse Kinematics, H-ANIM/MPEG4 Human Animation

1. Introduction

Motion capture is today the preferred approach to produce convincing human motions, especially those active motions involving interactions with the environment [14]. However, high production costs, low flexibility, and artifacts introduced by the capture process approximations have stimulated the proposal of numerous motion editing techniques. Some of them allow adjustments expressed in the posture space [22] or in the Cartesian space [6], or the retargetting to different characters [5, 10, 3, 15]. An extended discussion about the relative interest of preserving joint angles vs retaining Cartesian space constraints can be found in [19]. The continuity of the resulting movement being a key evaluation criteria, most of them work off-line as multiple pass editing tools. A minority of approaches target real-time retargetting for broadcast [19] or on-line applications [3]. In this latter context the continuity requirement is more difficult to enforce as only the past of the movement is known as opposed to the off-line context where all the movement information can be exploited. On the other hand on-line methods offer a great potential for the adaptive animation of autonomous characters moving in complex evolving contexts (e.g. in on-line games).

The motion editing approach presented here belongs to the per-frame family of methods as we want to exploit it for on-line adaptation of movement in the future. Presently the additional computing cost required for enforcing the prioritized constraint prevents its real-time use. It is exploited in a one-pass off-line context where the user predefines the timing and the relative priority of an arbitrary number of constraints.

The next sections develop how we improve the standard IK architecture to enforce multiple asynchronously overlapping Cartesian constraints at different priority levels while preserving the joint angles. Examples are given for a 3D humanoid (60 degrees of freedom) like the one performing a dance movement in Figure 1.

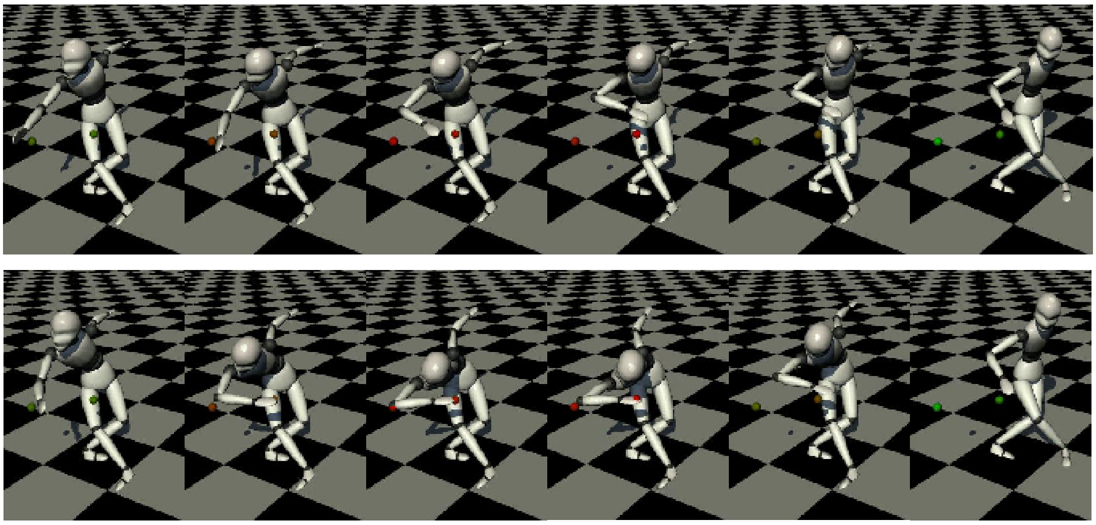


Figure 1: Example of motion editing with multiple prioritized constraints
(Top row: original motion, Bottom row: edited motion for arm and torso posture)

2. Motion Editing Strategy

The motion of a 3D character is usually represented as a set of joint trajectories together with the root trajectory in a world reference frame. This representation is very often used to measure the proximity of two motions[11]. However, a sufficient set of point trajectories is an equivalent representation as highlighted by recent discussions on motion metrics [8][9]. We want to stress that both representation capture valuable information for the end-user. The joint space representation captures the expressiveness of the motion; the joint states can be very easily mapped to characters complying with the same standard (like H-ANIM[7]). On the other hand the point trajectories capture the interactions with the external world [2]; for example the trajectory of points on the feet may indicate a path to follow, trajectory of points on the hand may indicate an object to grasp, other point trajectories can be important to express obstacle location, etc…

In the present approach, we exploit both motion representations to allow the end user to indicate *what* is important to preserve. The general strategy of motion editing consists in:

- Retaining the natural dynamics of the original movement by preserving:
    - The joint angle trajectories
    - The Cartesian trajectories of optional user-chosen points on the character.
- Adding user-defined Cartesian constraints wherever and whenever needed.

The association of a *priority level* to a constraint is the key element for ensuring a high flexibility to the motion editing process. This concept of *priority* should not be confused with the concept of *importance* dynamically evaluated in [19] in an on-line context. A high priority level strictly ensures the achievement of a constraint with respect to lower priority ones. On the contrary, an importance level is equivalent to a weight, thus leading to a compromise solution similar to the approach from Badler [24].

By default, the joint angle preservation is always requested and is assigned the lowest priority. This is a common aspect with the approach from [15]; the novelty of the present architecture comes from the exploitation of an arbitrary number of higher priority level constraints together with a smoothed goal management compatible with a one-pass processing.

3. Specifying the motion deformation

In our system the motion deformation is obtained through a set of constraints $\{c_k\}$ for $k=1,N$. Each constraint c_k consists at least in the definition of :

- An *effector* given by the position $\underline{\mathbf{p}}_{\mathbf{k}}$ of a point from the articulated structure expressed in the local frame of its parent joint (e.g. c1 and c2 in figure 2).
- A set of recruited joints $\{\theta_{\mathbf{kj}}\}$
- A *priority level*; for the sake of simplicity we suppose that each constraint is assigned a distinct priority level. This allows us to use the index **k** as the priority level, with $k=1$ being the highest level (think of it as a rank).

3.1. Favoring Synergies

A common approach in motion editing consists in partitioning the articulated structure into independent sub-structures that offer closed form solutions for user-defined constraints [25]. The drawback of this approach is the lack of global synergy in solving a set of conflicting constraints: some solutions might exist but the partitioning prevents their emergence. On the contrary, the present approach allows a constraint to recruit all or part of the joints from its parent joint up to the root joint. This stage of the constraint definition is called the *joint recruiting*; by default all the joints potentially influencing a constrained effector are recruited. Any subset of the default set is also allowed. The 6 dofs root joint is considered like any other joint at this stage. As a consequence, multiple constraints may compete for the control of some common joints. Very often the redundancy of the joint space allows to find solutions for all constraints. In case some constraints are conflicting, their distinct priority level sorts them in terms of constraint achievability. As a consequence the editing method can ensure the total achievement of the higher priority ones and the partial achievement of the lower priority ones.

The problem of overlapping joint chains was first described in [26]. However, instead of using an algorithmic scheme to solve this problem, we prefer to enforce a simple recruiting rule in order to obtain an effective enforcement of the scale of priority levels.

This rule is applied over the set of *potential common joints* $\{C_\theta\}$ as illustrated on Figure 2 for two constraints c_1 and c_2. Let $\{\theta_{kj}\}_C$ indicate the intersection of the recruited joint set $\{\theta_{kj}\}$ from constraint c_k with the set of potential common joints $\{C_\theta\}$. The rule states that the recruiting of low priority constraints must be included into the recruiting of high priority constraint, over the common joint set :

$$\forall k \geq k_0, \{\theta_{kj}\}_C \subset \{\theta_{k_0 j}\}_C$$

Failing to do so violates the hierarchy of priority levels because a low priority constraint can animate a part of the articulated structure for which a dependent high-level constraint has no control.

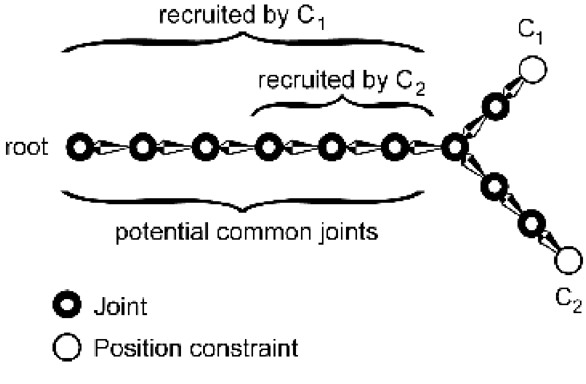


Figure 2: Intersection of the recruited joint sets of constraints c_1 and c_2 over the *potential common joints* $\{C_\theta\}$; in the present case the priority of c_1 (k =1) is higher than the priority of c_2 (k =2).

3.2. Effector trajectory management

Let us denote the *original motion* of the character as $\{\theta_{af}\}$ with $1 \le a \le N_a$ (number of articulations), and $1 \le f \le N_f$ (number of frames). The point effector $\underline{\mathbf{p}}_{\mathbf{k}}$ associated to the constraint $\boldsymbol{c_k}$ has a corresponding *original trajectory* in the world frame, noted $\{\underline{\mathbf{P}}_{\mathbf{kf}}\}$, $1 \le f \le N_f$. Figure 3 highlights the original trajectory of the chain tip effector.

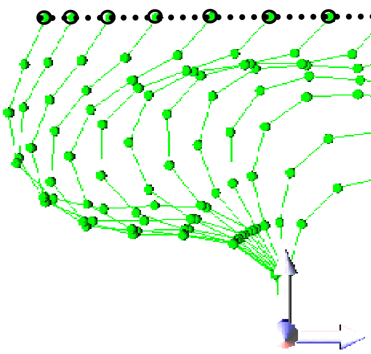


Figure 3: Sampled motion of an articulated chain highlighting the *original trajectory* of the chain tip effector (time flows from left to right)

For each constraint $\boldsymbol{c_k}$, the user can define a set of goals $\{\mathbf{g}_{\mathbf{ki}}\}$, $1 \le i \le N_g$ (number of goals). In case the goal set is empty, the constraint's effector is attracted towards its original trajectory $\{\underline{\mathbf{P}}_{\mathbf{kf}}\}$. The interest of this default behavior is to allow the user preserving any Cartesian constraint of interest from the original motion when other constraints deform the motion.

Each goal $\mathbf{g}_{\mathbf{ki}}$ has an *activity timing* in order to merge it smoothly with the effector original trajectory. The *activity* is a normalized variable varying between 0 (inactive) and 1 (full activity). The activity timing consists in four consecutive key times $\{\mathbf{t_0}, \mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}\}$ set within the motion duration $[\mathbf{t_I}, \mathbf{t_F}]$. Figure 4 shows the corresponding activity evolution $\mathbf{a}_{\mathbf{ki}}$ between [0., 1.].

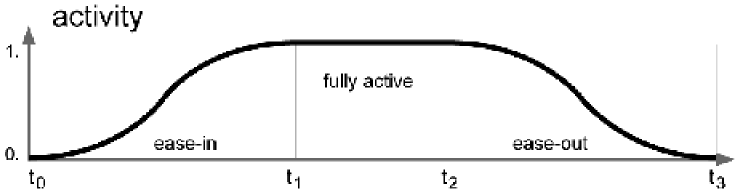
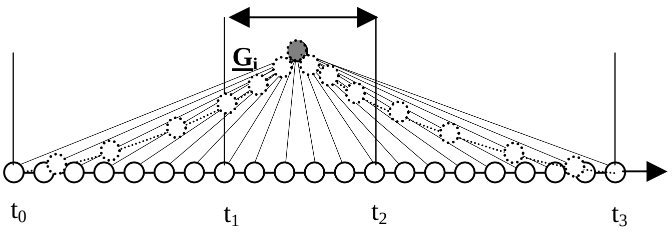


Figure 4: Default activity profile with cubic steps for ease-in and ease-out. The key times can be arbitrarily set over the original motion duration $[t_I, t_F]$.

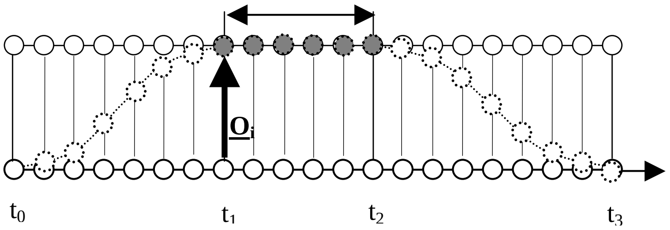
The deformation specified by a goal can have one of the two following forms:

- **A position goal $\mathbf{G}_i$** attracts the effector while the goal is active. Figure 5a illustrates such a case where the gray dot is the position goal and the bottom line is the original trajectory of the effector. The activity timing is the one from Figure 4 resulting in the construction of in-between goals shown as white dotted points (cf section 3.3 for their construction).
- **An offset vector $\mathbf{O}_i$** serves to define a *goal trajectory* by offsetting the original trajectory over the activity timing. Figure 5b illustrates this context with the timing from Figure 4. As this approach produces an overall smoother trajectory, it is the default mode proposed to the user. (cf section 3.3 for its construction).

Another interesting option for the position goal type is the possibility to attach it to any frame of the articulated structure (instead of having a fixed goal in the world coordinate system). This helps to constrain relative positions of body parts, e.g. the hand and the face of a humanoid, even if the motion is extensively transformed by other constraints.



a) Attracting the effector towards a fixed point $\mathbf{G}_i$



b) Attracting the effector towards a goal trajectory obtained by offsetting the original trajectory with $\mathbf{O_i}$

Figure 5: Comparison of the resulting trajectories obtained with the two types of goals
(The double-edge arrows indicate the interval of full activity from Figure 4; time flows from left to right).

3.3. Composite goal construction

For each frame ***f***, a constraint $\boldsymbol{c_k}$ needs to know the position of its *composite goa*l $\underline{\mathbf{P}}_{\mathbf{kC}}$. The composite goal is constructed from the knowledge of two sources:

- Its goal set $\{\mathbf{g_{ki}}\}$
- Its effector position from the original trajectory $\{\underline{\mathbf{P}}_{\mathbf{k\,f}}\}$

Each source builds a desired position for the effector and a weighted average defines the composite goal as follow:

$$\underline{P}_{kC} = \max(a_i).\frac{1}{\sum_i a_i}.\sum_{i=1}^{N_g} a_i \underline{P}_{ki} + (1-\max(a_i)).\underline{P}_{kf}$$

In the first term we find the center of mass of the positions $\underline{\mathbf{P}}_{\mathbf{ki}}$ proposed by the goals according to their mode and weighted by their current activity $\boldsymbol{a_i}$. The first term is also weighted by the maximum of the goal activities $max(a_i)$ to smoothly mix it with the second term attracting the goal towards the original trajectory. This approach guarantees the smoothness of the composite goal trajectory, especially in the transition phases at the beginning and the end of the goal activation.

3.4. Allowing multiple asynchronous constraints

An additional level of composition appears when two or more constraints overlap partially in time. This is necessary when different parts of the articulated structure have to be constrained. The concept of *priority* is exploited within the IK solver (section 4) to obtain a strict hierarchy of constraints achievement [1]. The prioritized constraints and the smooth composite goal trajectory ensure that asynchronous overlapping constraints can be treated in a one-pass process without introducing discontinuities at their activation-deactivation transitions.

4. Enforcing the motion deformation with IK

We first describe the general structure of the motion deformation algorithm prior to briefly describe how a Jacobian-based IK can enforce multiple levels of priority among the constraints.

4.1. General architecture for motion editing

The pseudo-code of the prioritized constraint-driven motion deformation with IK is listed in the figure 6 below.

```
Initialization
For each frame f, from 1 to $N_f$
  Compute absolute constraints goals
  Do
    Compute relative constraints goals
    Compute joint angle preservation vector
    Run one IK step [1]
  While (halting criteria not met)
  Store resulting posture
End for
```

Figure 6: Pseudo-code of the motion editing algorithm

The initialization phase builds the data structures of the constraints including the memory space allocation for their Jacobians and for the priority-enforcing projection matrices. It pre-computes and stores also the effectors' original trajectory. Then the main motion deformation loop iterates through the sampled frames of the original motion. Composite goals (see 3.3) and the joint angle preservation are evaluated and fed to the IK engine which runs one convergence step. The convergences is stopped if one of the following condition is met:

- Number of iterations runs over an upper limit
- The *variation* of the norm of the constraints errors and the variation of the norm of the joint angle preservation error are under respective thresholds. Testing the variation rather than the absolute value of the errors is necessary as lower priority constraints, and especially the joint angle preservation, have a smaller or even no solution space to be realized.

4.2. Ensuring the strict priority levels

Like the classic numeric IK approach, we rely on the Jacobian matrix gathering the partial derivatives of the constraints variables with respect to the joint parameters. Building a position Jacobian like the ones we exploit is straightforward [21]. Solving for multiple constraints depending on a common set of joints requires considering them in the same linearized system. If we gather all the constraints Jacobians by piling them into a unique one, we end up with a compromise solution [16, 23]. We compute instead a specific Jacobian for each priority level. In order to enforce distinct priority levels, we first project each Jacobian onto the *null-space* of the augmented Jacobian of all the higher priority levels. Doing so allows us to ensure that a solution will not perturb the solutions of higher priority levels. The reader could refer to [1] to find a more detailed explanation on the architecture of prioritized Inverse Kinematics.

Once the hierarchy of prioritized kinematic constraints has been taken into account there might remain a non empty solution space for the optimization of cost functions directly expressed in the joint space. This is generally the case as the joint space dimension is usually far greater than the dimension of all the kinematic constraints. This is where we deal with the preservation of the joint angle trajectories defined by the original motion. By default our joint angle preservation term is proportional to the difference between the current joint value and the original motion value. It can be easily shown that such a term is the gradient of a cost function expressing the distance to the posture of the original motion.

Another approach exploited in on-line motion retargetting in [3] aims at minimizing the difference to the original joint velocity. Such cost functions and many others expressed in the joint space can be weighted and combined to achieve specific effects in the original motion tracking (see [4] for a list of such cost functions in Robotics).

5. Results

We have tested our prioritized motion editing method in several configurations, three of which are detailed in this section. However, we do not provide any computational time, as we strongly believe that the efficiency of our IK solver, which is the bottleneck of our algorithm, could be greatly enhanced by finely tuning specific convergence parameters. Moreover, precisely defining the halting criteria should greatly reduce the number of iterations performed at each frames by the IK solver. Thus, we consider that the computational time presently obtained does not represent the potential efficiency of our method and should be greatly reduced in the near future.

In the first example (Figure 1), a high priority constraint is assigned to the right elbow while a middle priority constraint adjusts the posture of the forearm while the joint angle preservation acts at the lowest priority level. The joint recruiting includes all the joints from the wrist to the spine joints. The wrist and elbow are conflicting to control the arm but as the elbow has a higher priority the wrist constraint still serves to orientate the forearm in the desired direction. Finally, as the joint angle preservation is continuously active, the transition with the original movement is very smooth.

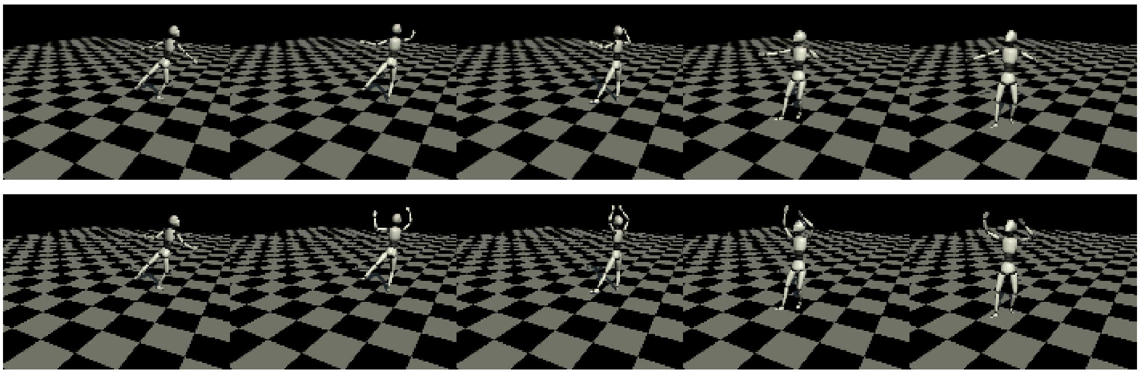


Figure 7: An original motion (top) edited to raise the arms with prioritized constraints on the wrists and the elbows (bottom).

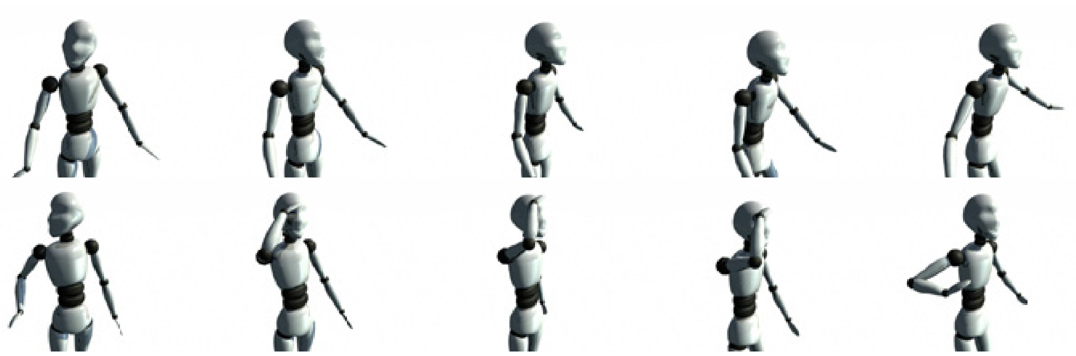


Figure 8: Original motion (top) edited to bring the hand to a location expressed in the head coordinate system (bottom).

In the second example (Figure 7), the same scale of constraints and relative priorities is used to change the posture of both arms so that they adopt a classic posture while the character is moving. Adjusting the constraint is easy as the goals are expressed in the local coordinate system of the thorax. Although the posture difference is significant, the resulting motion remain continuous during the transition phases.

In the third example (Figure 8), the goal of the constraint is expressed in the local frame of the head, thus allowing the hand to easily express the desired “gazing with sunshine protection” posture while the head turns.

6. Discussion and conclusion

Despites its higher computing cost compared to solutions based on analytic IK solution our prioritized IK scheme offer the following strong points from the end-user point of view:

- Specification of priority level for conflicting constraints
- Possibility to handle asynchronous overlapping constraints
- Customization of the joint recruiting
- Goals relative to the original movement or to the current posture (in addition to traditional absolute goals in position and orientation)

Together with the smooth composite goal construction, the prioritized IK guarantees that successive overlapping constraints can be treated in a one-pass process without introducing discontinuities at the goal level. In addition, the joint angles preservation greatly helps retaining the continuity of the original movement. Future work will first refine the management of the IK convergence loop by exploring the use of continuity-enforcing cost functions in the posture space. Indeed, instead of adding a post processing stage to filter the results but altering the solution, we believe that we can easily exploit the priority levels of our IK solver to define hard-constraints limiting the norm of the changes between neighboring frames and thus enforcing the continuity of the final motion. Doing so integrates the treatment of the continuity problem directly at the IK solver. This should have a positive impact on the computing cost, which is too large presently. Other parameters of the IK solver as the convergence step and the damping factor will be tuned too.

On the user side, we target the integration of constraints on the center of mass as it is generally an important characteristic of full body movements [12, 17, 20]. As such it will be possible to assign a high priority to its preservation or editing.

Acknowledgements

This research has been supported by the FNRS under the grant N° 2000-061999.00. The final rendered animation sequences have been obtained with Maya through the Alias/Wavefront Research Donation program. We also wish to thank Nicolas Elsig for the design of the humanoid character .

References

1. Baerlocher P., Boulic R. **Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures.** Proc. of IEEE IROS 98, Victoria, pp. 323-329, Oct. 1998
2. Bindiganavale R., Badler N. **Motion Abstraction and Mapping with Spatial Constraints.** Proc of Captech98, LNAI 1537, pp 70-82, Springer-Verlag Berlin Heidelberg
3. Choi K.J, Ko H.S. **On-line Motion Retargeting**. Journal of Visualization and Computer Animation 11(5): 223-235 (2000)
4. Cleary K., Tesar D. **Incorporating Multiple Criteria in the Operation of Redundant Manipulators.** Proc. of IEEE Conf. of Robotics and Automation, Vol 1: 618-623 (1990)
5. Gleicher M. **Retargetting Motion to New Characters.** Proc. of SIGGRAPH'98, pp 33-42.
6. Gleicher M. and Litwinowicz P. **Constraint-based Motion Adaptation.** The Journal of Visualization and Computer Animation, 9, 65-94 (1998)

7. H-ANIM, Humanoid Animation Working Group, www.hanim.org
8. Kovar L., Gleicher M., Pighin F. **Motion Graphs.** Proc. of SIGGRAPH’2002, Austin
9. Lamouret A., Van De Panne M. **Motion Synthesis by Example**, Proc. of the Eurographics Workshop on Computer Animation and Simulation, EGCAS’96, Poitier, Eds Boulic and Hegron, ISBN 3-211-82885-0, 1996
10. Lee J. and Shin S.Y. **A Hierarchical Approach to Interactive Motion Editing for Human-Like Figures**, Proc. of SIGGRAPH'99, Los Angeles
11. Lee J., Chai J., Reitsma P.S.A., Hodgins J., Pollard N.S. **Interactive Control of Avatars Animated with Human Motion Data**, Proc. of SIGGRAPH’2002, Austin.
12. Liu C.K., Popovic Z. **Synthesis of Complex Dynamic Character Motion for Simple Animation**, Proc. of SIGGRAPH’02, Austin
13. Maciejewski A.A. **Dealing with the ill-Conditioned Equations of Motion for Articulated Figures.** IEEE CGA, Vol. 10, n°3, pp 63-71, 1990
14. Menache A., **Understanding Motion Capture for Computer Animation and Video Games.** ISBN 0-12-490630-3, Morgan Kaufmann, 2000
15. Monzani J.S., Baerlocher P., Boulic R., Thalmann D. **Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting.** Proc. Eurographics 2000, Interlaken August 2000
16. Philips C.B., Zhao J., Badler N.I. **Interactive Real-Time Articulated Figure Manipulation Using Multiple Kinematic Constraints.** Computer Graphics 24 (2),pp 245-250, 1990
17. Popovic Z. and Witkin **A. Physically Based Motion Transformation.** Proc. of SIGGRAPH'99, Los Angeles, August 8-13, 1999
18. Press W., Teukolsky S., Vetterling W.T., FLANNERY B.P. **Numerical Recipes in C.** second edition, Cambridge University Press, ISBN 0 521 43108 5, pp59-70, 1992
19. Shin H.J., Lee J., Shin S.Y., Gleicher M., **Computer Puppetry: An Importance-Based Approach.** ACM Transaction on Graphics, Vol 20(2), April 2001, pp 67-94
20. Tak S., Song O.Y., Ko H.S. **Motion Balance Filtering.** Proc. of Eurographics’2000, Computer Graphics Forum, Vol 19(3), Blakwell publishers 2000
21. Watt A., Watt M. **Advanced Animation and Rendering Techniques.** Addison-Wesley, ACM Press, 1992
22. Witkin A., Popovic Z. **Motion Warping.** Proc. of SIGGRAPH’95, Los Angeles, 1995
23. Zhao J., Badler N. **Inverse Kinematics Positioning using Nonlinear Programming for Highly Articulated Figures.** ACM Transactions on Graphics, Vol. 13, No. 4, pp. 313 - 336, Oct. 1994
24. Badler N.I., Manoochehri K.H., Walters G. **Articulated Figure Positioning by Multiple Constraints.** IEEE Computer Graphics & Applications, Vol. 7(6), pp. 28 - 38, June 1987
25. Tolani D., Goswami A., Badler N.I. **Real-time inverse kinematics techniques for anthropomorphic arms.** Graphical Models, Vol. 62, pp. 353 - 388, 2000
26. Badler N. I., O'Rourke J. and Kaufman G., **Special Problems in Human Movement Simulation,** In proceedings of ACM SIGGRAPH, Annual Conference Series, pp. 189-197, 1980.