Bringing the Human Arm Reachable Space to a Virtual Environment for its Analysis

I. Rodríguez[a], M. Peinado[a], R. Boulic[b], D. Meziat[a]

[a]Computer Engineering Department, University of Alcalá, Madrid, Spain
[b]Swiss Federal Institute of Technology, Lausanne, Switzerland

inma@aut.uah.es, manupg@aut.uah.es, ronan.boulic@epfl.ch, meziat@aut.uah.es

Abstract

Human reaching in a 3D environment is an interesting matter of research due to its application to workplace or vehicle-interior design. We introduce a 3D environment where a virtual human performs reaching tasks over 3D objects in the world. This environment also provides tools to generate and visualize reachable volumes. Reachable spaces are approximated using adjacent box-shaped voxels. We define several strategies in order to model different types of reaching, and we employ our system to construct and analyze reachable spaces for these strategies. In general, different strategies will have reachable spaces that share a common region of intersection. Therefore, goals will exist that can be reached using two or more strategies. For those goals, a high-level layer is responsible for selecting the most appropriate given a certain reaching task. As a practical application, this paper presents a comparison of two usual strategies to model standing and seated reaching. The generated reach spaces show that, for each of them, a strategy is clearly more adequate than the other.

1. Introduction

The study of reaching tasks in humans has a number of important applications. For example, workplace designers are in great need of reachability information, in order to develop working places where everything is easily accessible, thus reducing the risk of musculo-skeletal damage. A similar reasoning applies to vehicle-interior design. The designer of a plane's cockpit, for instance, must place all instruments and displays carefully in order to keep them reachable at any time.

The research we present here is aimed at simulating reaching motions of animated characters in a virtual environment. It also helps studying human reachability by allowing for the interactive manipulation and visualization of reachable spaces.

Our approach is high-level, which ensures that the management of virtual humans and the creation of character animations are managed in an effective way. We focus exclusively on modeling hand reaching, but our system could also be adapted to model the reach of the foot, the knee or other body parts.

2. Related Work

Reaching tasks have been an object of study in ergonomic researches; the main reason is that a worker's workplace has to be well designed in order to prevent postural problems [1].

The comprehensive book by Badler et al. includes a discussion on reachable space determination using inverse kinematics [2]. This multi-joint simulational approach, which we also use in our work, is opposed to the one found in researches like [3], that compute reachable volumes analytically and are limited to structures with a few degrees of freedom (two in the mentioned work). A robotic study also evaluated a kinematics chain in closed form [4]; a method was developed for delineating surface patches defining the reach envelop of the chain.

A research produced natural reach postures by interpolating motion captured data [5]. A more recent work has proposed a method to predict reach motions based on experimental data. A functional regression analysis was utilized to model how joint angles change over time [6]. Mas discerned three reach areas depending on the distance from the hand to the target [7]. He defined different strategies for each area considering the need for controlling the center of mass or for additional supports.

3. Construction of Human Reachable Volumes

Our objective is to know what points in 3D space are reachable by a virtual human following a determined reaching strategy (direct, crouch…). To do that, we propose a data structure that is especially suitable to approximate certain types of volumes. It is called *Volume Approximation Tree* (VATree), and basically consists of an octal tree of box-shaped voxels, or boxes. We say it is an octal tree because after every subdivision each parent nodes is split into eight children. In the literature, octal trees have been utilized as an efficient way to represent volumes or surfaces [8]. Once a VATree has been generated, it contains reachability information for a certain strategy, and can be queried in a later stage to determine whether a point is reachable using that strategy.

The process of generating a VATree starts with an initial reach box. The resulting VATree will be fully contained inside the initial box, so this must be chosen carefully. For example, Figure 1 (left) shows a virtual human situated in front of the initial box that will be used to generate his frontal reachable space. Figure 1 (right) shows the same virtual human and an initial box that will be used to obtain his frontal reachable space in a sitting posture. In the latter case, the initial box envelops the surface of the table whose reachability is under study.

The animation technique used in the generation of the reachability trees is Inverse Kinematics (called IK in the following). This technique lets the user attach Cartesian constraints (goals) to some parts of the human body (effectors) [9][10], then the system automatically calculates the joint angles needed to reach the goal.

Figure 2 illustrates the process of constructing a VATree. It also depicts the interaction between our reachability and IK modules. The construction process begins with an initial box in which the approximated volume will be contained. The IK module, in order to decide whether a point is reachable, needs three inputs: an effector, a goal and some constraints that characterize the reaching task.

In an iterative process, these three inputs are assigned as follows: the goal is sequentially taken to be each vertex of the initial box; the effector is the part of the body that does the reaching; and the tasks define the type of reaching, i.e. controlling the center of mass, establishing the motion-flow root, flexing the legs, etc.

If the eight vertices defining the box are reachable, then it is assumed that all the points inside that box are reachable. If, on the contrary, all vertices are unreachable, then the whole box it considered unreachable, so the space inside the box will not be a part of the generated volume. An intermediate situation occurs when some of the vertices of the box are reachable while others are not. In that case, the box is divided in eight smaller boxes of equal size, and the process is repeated for each of this new boxes. The tree is considered finished when its maximum depth has been attained.

Note that the voxels that form the tree are adjacent, and therefore every vertex is shared by several of them. This implies that performing eight IK simulations –one at each vertex– for every box is extremely redundant. Our actual algorithm eliminates this redundancy by performing exactly one simulation at each vertex.

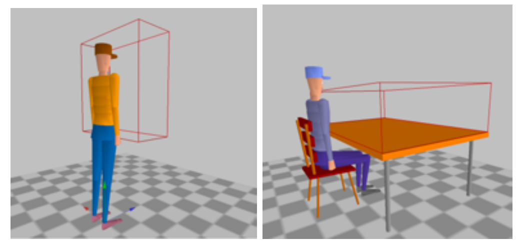

Figure 1. Two examples of initial boxes

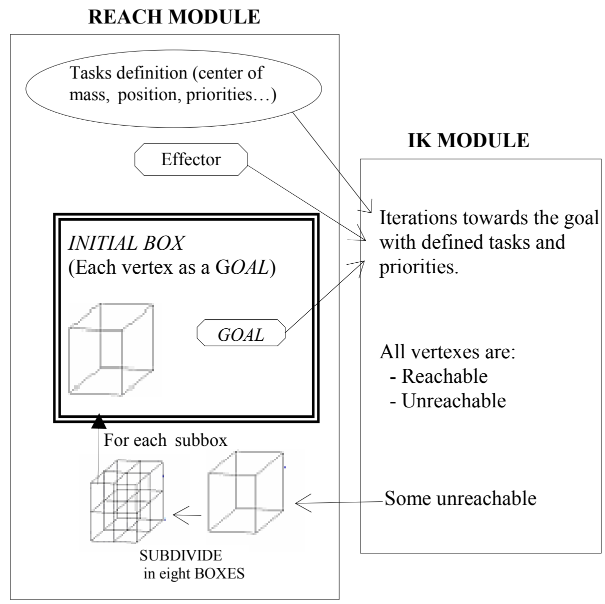

Figure 2. Interaction between Reach and IK modules

4. Managing Reaching Strategies in a Virtual Environment

When faced with a reaching task, an individual may adopt several reaching strategies. If the goal is near enough a direct reaching is possible, in which the subject simply stretches his arm. Also, an additional pelvic movement may be necessary in order to ensure balance. A slightly more involved situation occurs for goals outside the so called *near-reach area*. For instance, if the goal is in a low position the subject may decide to adopt a crouch posture, or even to take a step forward to facilitate the reaching motion. As a matter of fact, different strategies result in different types of reaching. For every reaching task, characterized by a goal in space, there is an optimal strategy which will depend, among other things, on the initial posture of the individual.

Figures 3 and 4 below are provided for a visual comparison of two common hand reaching strategies. In the left one, the goal lies inside the reachable space of the direct strategy, and thus this strategy is adopted. In Figure

4, on the other hand, the goal is so low that lies outside the reachable space of the direct strategy. Therefore, the subject employs a *crouch* strategy that, thanks to a knee-bending movement, makes the reach possible.

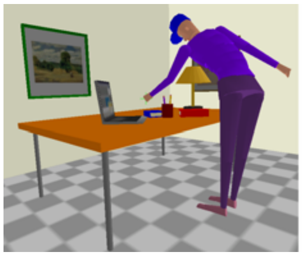

Figure 3. Direct Strategy

Figure 4. Crouch strategy

The above discussion implies that a mechanism is needed to efficiently store and manipulate reachable spaces. Our choice, as explained in Section 3, is to represent reachable spaces as a set of box-shaped adjacent voxels. Figure 5 shows the reachable spaces for some of the available strategies, namely for those in relation to standing reaching. All volumes shown correspond to left hand reaching.

In general, as seen in Fig. 5, different strategies will have reachable spaces that share a common region of intersection. As a result, goals will exist that can be reached using two or more strategies. For those goals, a high-level layer is responsible for selecting which, among those strategies valid to reach the goal, is the most appropriate one for a given reaching task. The decision is based on a priority scheme, which works as follows.

Higher priorities are assigned to strategies that involve a smaller cost from a kinematic point of view. According to this, wherever two or more strategies conflict, the one with the highest priority is selected. For instance, it is obvious that the direct/crouch strategies introduced above do share a significant region of intersection. Since the direct strategy does not require knee bending, it is assigned a higher priority, and thus it will be selected for any goals located in the region of intersection.

The VATree data structure we propose offers an adequate means to find whether a goal is reachable with a certain strategy, since its recursive nature allows for fast containment tests (i.e., tests that determine if a point is inside or outside the represented volume).

Our system manages a VATree for each available strategy, and by querying those trees it can select the strategy that is most suitable for a given task. However, VATrees provide only information about reachability. This means that, given a goal in space, a VATree tells only whether its associated strategy is valid to reach the goal. Therefore, there is still a need for postural information: which posture, i.e., which combination of joint values, makes the reach possible.

There are several ways to address this problem. Ours consists in performing inverse kinematics (IK) simulations at two different stages: first, VATrees are computed by recursively subdividing space (see Section 3) and performing IK simulations to determine the reachability of the resulting voxels. On a second stage, our system utilizes the previously generated VATrees to evaluate the reachability of given goals, and performs additional IK simulations that yield the definite reach postures.

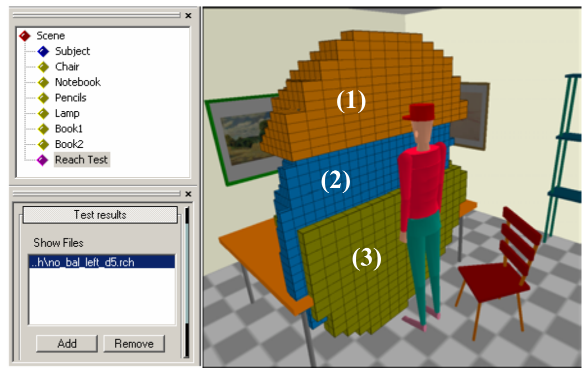

Figure 5. Virtual Environment showing the reachable spaces of some standing reaching strategies: tip-toe (marked with a 1), direct upper-body (2) and crouch (3)

Figure 6a shows the reachable spaces of two common standing reaching strategies: *Direct Upper-Body Reaching* (DUB) and *Direct Integral Reaching* (DI). In DUB, the IK simulation only affects the upper body — not including the pelvis. This is opposed to DI, in which the whole hierarchy takes part in the motion, thus yielding postures that reach more distant points. This is due to the additional mobility provided by the hip, knee and ankle joints, which play a key role when reaching further goals, since they allow for a backward pelvic movement which is essential for balance maintenance.

Our system supports both DUB and DI strategies, with the former being assigned a higher priority so that DI, which has a bigger cost, is only chosen for those goals that are not reachable by DUB —this tries to imitate real life, in which one seldom uses the lower body in a reaching motion unless the goal can't be reached otherwise.

Figure 6b shows a similar comparison for the seated reaching case. Two strategies are compared: *Naive Seated Reaching* (left) and *Normal Seated Reaching*. The difference between them lies in which parts of the body they affect. With the naive strategy, only the upper body participates in the motion, much like in the DUB strategy presented above. With the normal strategy, on the other hand, the hip joint adds an extra degree of mobility, giving rise to more natural postures that, in addition, facilitate to reach further.

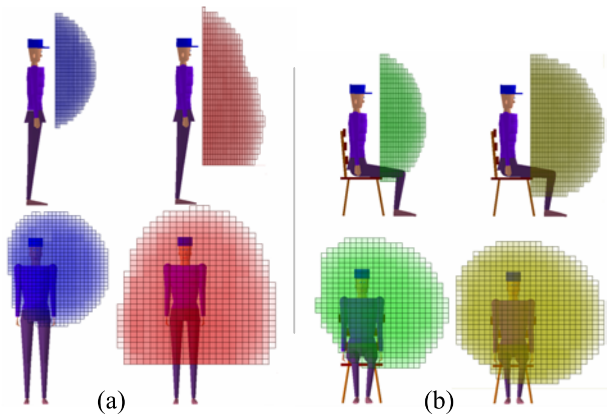

Figure 6. Comparison of reachable spaces for different reching strategies. (a) Standing reaching with DUB (left) and DIR strategies. (b) Seated reaching with Naive (left) and Normal strategies.

5. Conclusions and Future Work

We have introduced a system that helps studying human reachability. Given a certain reaching task, characterized by a goal in space, an initial posture and by the part of the body which has to reach (i.e., the left hand), our system can decide which strategy, among those available, is the most suitable to successfully complete the task. VATrees are a key feature in our analysis of reach. They provide an efficient way to store and manipulate the reachable spaces of the different strategies. Our system integrates the functionality to generate this VATrees and to exploit them, once generated, in order to choose the most suitable strategies for a given reaching task.

This system is not limited to the predefined set of strategies. It can be extended with any strategy that one could devise. At the moment several strategies are available, both for standing (*DI*, *DUB*, *Crouch*, *Tip-toe*…) and seated reaching (*Naive, Normal*…). However, there is plenty of room for new strategies that contribute to improve the realism of the resulting postures. For instance, a strategy that we plan to implement soon is *Upper-Body Torsion*, which will permit reaching goals placed on the sides of the subject.

Several issues remain to be addressed regarding many aspects of our system. One of them is self-collision detection. It is not taken into account at the moment, but should be in order to detect, and possibly correct, postures that are invalid for containing self-collisions.

Another issue we intend to focus on is joint fatigue and its effect on the reachable spaces. We have already developed methods to asses fatigue at joint level [11], and we plan to employ them to study the relation between arm fatigue and reachability.

Finally, one question we also consider worth exploring is that of obstacle presence and its impact on the generated reachable spaces. This could have interesting applications in the fields of workplace and vehicle design, since it would allow for an optimal placement of objects, from the point of view of reachable space maximization.

6. References

[1] Eui S. Jung, Dohyung Kee and Min K. Chung, *Reach Posture Prediction of Upper Limb […] for Ergonomic Workspace Evaluation*, Proceedings of the Human Factors Society.-36th Annual .Meeting, 702-706, 1992.

[2] N.I. Badler, C. B. Phillips, B.L. Webber. *Simulating Humans: Computer Graphics, Animation, and Control,* Oxford University Press, 1993.

[3] Kee, D.H., Shin, Y.T., Kang, D.S., and Chung, E.S. *Reach Volume*. Proceedings of the Fall Conferences of Korean Institute of Industrial Engineers, 232-237, 1996.

[4] Abdel-Malek, K. and Yeh, H.J., *Analytic Boundary of the Workspace for General Three Degree of Freedom Mechanisms.* International Journal of Robotic Research. Vol. 16, No 2 , 198-213, 1997.

[5] D.J. Wiley, J.K. Hahn. *Interpolation Synthesis of Articulated Figure Motion*. IEEE Computer Graphics & Applications, Vol. 17, No 6, Nov./Dec. 1997.

[6] J.J. Faraway, *Modeling Reach Motions Using Functional Regression Analysis*. Digital Human Modeling for Design and Engineering Conference and Exposition, Michigan, June. 2000.

[7] R. Mas, R. Boulic, D. Thalmann. *Extended Grasping Behavior for Autonomous Human Agents*, ACM Conference on Autonomous Agents'97, Los Angeles. 1997.

[8] D. Libes. *Modeling Dynamic Surfaces with Octrees*, Computers & Graphics Magazine, Vol. 15, No 3, 1991

[9] Baerlocher P., Boulic R., *Task-priority Formulations for the Kinematic Control of Highly Redundant Artigulated Structures*, Proc. of IROS'98, Victoria, Canada, Oct. 1998.

[10] R. Boulic, R. Mas, D. Thalmann, *A robust Approach for the Center of Mass Position Control with Inverse Kinetics*. Journal of Computers and Graphics'96, Vol. 20, Nº 5. 693-701, 1996.

[11] I. Rodríguez, R. Boulic., D. Meziat. *A Joint Level Model of Fatigue for the Postural Control of Virtual Humans* Proceedings of Human and Computer 2002, 220-225, Tokyo, 2002.