Adapting Behaviors to New Environments, Characters, and Tasks

Nancy S. Pollard and Jessica K. Hodgins

College of Computing and Graphics, Visualization and Usability Center*

Abstract

Animated characters and robots are exhibiting increasing levels of competence in complex tasks ranging from locomotion to juggling. One continuing challenge, however, is to develop behaviors that are robust with respect to variation in the environment, the character, and the task. Examples of a behavior (e.g. an individual performing a motion) are a rich source of information, but reliable techniques are needed to adapt examples to new situations. This paper describes how task information can be used to adapt existing examples to changes in the environment or to changes in the character's physical characteristics. Sets of rules for adapting or scaling examples to new situations allow a behavior to be described in a compact way, while also capturing some of the benefits of the information that may be contained in the examples. Results are shown for adapting example grasps to new object geometries and adapting dynamic behaviors to characters with differing physical characteristics.

1 Introduction

Bernstein [3] describes dexterity as "the ability to find a motor solution for any external situation, that is, to adequately solve any emerging motor problem correctly, quickly, rationally, and resourcefully." Dexterity in this sense is needed for autonomous characters or robots that will be placed in unpredictable environments. For example, we might want to create a robot that can gracefully manipulate any object in a hardware store, or we might want to create virtual characters that can "act" out a crowd scene in response to high level instructions from a user.

Creating animated characters or robots that have a high level of dexterity has proven to be extremely difficult. Achieving this goal will require developing behaviors that are robust with respect to variation in properties of the environment, the character, and the task. Variation in the environment includes obstacles that a character must avoid, or a variety of objects to be manipulated. Variation in the properties of the character can range from simple modifications (e.g. for carrying a load) to a complete character redesign. For example, an animator may want to design a new character and use an existing set of behaviors to control that new character. Finally, variation in the task can include any variations that may be requested by a user. For example, a user may want their character to run faster or to dribble a basketball closer to the ground.

One appealing way to create behaviors is to make use of examples. Examples are rich sources of information, and may include many subtleties that we do not know how to model in a compact manner. Example data collected by measuring an actor's performance, for instance, can capture an actor's strategy for performing a particular task. It can also capture that actor's style.

If examples are to be used as part of a behavior definition, we must have a means of selecting appropriate examples and tailoring them to fit each new situation that arises. One approach to this problem is to obtain a set of examples that completely spans the behavior space. In this case, relatively simple functions may be sufficient for interpolating between examples [26] [1] [29].

We may not always be able to generate or store a sufficiently large set of examples to obtain good results from simple interpolation functions, however. This paper describes how information about a behavior or task can be used to adapt existing examples to changes in the environment or to changes in a character's physical characteristics. Sets of rules for adapting or scaling examples to new situations allow a behavior to be described in a compact way, while also capturing some of the benefits of the rich information that may be contained in the examples. Section 2 outlines some previous research in creating behaviors that can adapt to changing situations, Section 3 describes a technique that uses a static force/torque analysis to adapt grasps to new object geometries while ensuring that the new grasp meets the constraints of the task, Section 4 describes a technique that uses a dynamic analysis to scale behaviors such as running to work with characters having different physical characteristics, and Section 5 presents a discussion of the results.

2 Background

A number of different strategies have been used in the robotics and graphics communities to achieve robust behavior in response to changes in the environment, physical characteristics, or task. At a high level, these strategies can be divided into procedural, planning, and example-based approaches.

In the area of procedural approaches, Brooks [5] de-

*Georgia Institute of Technology, Atlanta, GA 30332-0280, [nsp | jkh]@cc.gatech.edu

scribes an architecture for constructing complex, reactive behaviors. A similar approach has been used by a number of researchers for creating autonomous animated creatures: Reynolds [24] developed flocking behaviors for groups of animated creatures; Blumberg and Galyean [4] describe a virtual creature that can either behave autonomously or be directed by a user; Tu and Terzopoulos [27] describe a set of integrated behaviors for artificial fish; and Perlin and Goldberg [20] describe a system for creating actors with distinct personalities that respond to users and to each other in a virtual environment. Dynamically simulated behaviors for human motion [12] [32] [6] may function well over a wide range of user-specified parameters such as running velocity or jumping height, and behaviors such as balancing may be designed to be robust to unexpected disturbances. In general, procedural approaches can result in impressive behaviors and may allow a character to react quickly to changing situations. The disadvantage of this type of approach is that the burden rests with the designer of a behavior to create a plausible response to every situation that is likely to be encountered.

An alternative approach to generating robust behaviors involves planning each new action based on the current situation, a set of constraints, and an evaluation function. This approach has been used widely for planning motions for robotics [15], and it has been applied to planning manipulation behaviors for animation [14]. Also in the animation domain, trajectory optimization approaches [30] [17] [33] allow an animator to specify constraints on a motion (including key poses such as an initial and final pose) and an evaluation function such as minimum energy, and then compute the motion that minimizes the objective function while meeting the specified constraints. This approach allows an "optimal" response to be computed for any situation, but a set of constraints and an optimization function must be defined to completely describe a behavior. This can be a difficult task for complex, coordinated behaviors such as walking or running.

A third approach to generating robust behaviors involves interpolating between or extrapolating from existing examples of a behavior. A large amount of research has been done on memory-based or nonparametric learning techniques [26] that compute a mapping from input data to output data based on a set of stored examples. Atkeson, Moore, and Schaal [1] review how memory-based learning has been used for motion control in robotics. In animation, Wiley and Hahn [29] have used interpolation between motion examples to generate reaching motions for an animated character. Unuma, Anjyo, and Takeuchi [28] and Bruderlin and Williams [7] use interpolation in the frequency domain to allow an animator to blend between different sets of experimental motion data to produce walking, running, and gestural motion with different styles.

When animated characters or robots have a large number of degrees of freedom, it can be difficult to obtain a sufficient number of examples to use simple interpolation techniques. If a given set of examples does not adequately span the space, simple interpolation may result in motion that violates task constraints. For example, interpolated motion for an animated character may not result in a convincing motion to grasp or manipulate an external object, and interpolated control parameters for a dynamically simulated character may not result in a running character that can maintain its balance. Because of this difficulty, it is important to understand how much information can be extracted from a single example. In previous work, Atkeson and Schaal [2] describe learning a pendulum swing-up task for a Sarcos robot arm based on a human demonstration of the same task; they also review related work on imitation learning for automatic robot programming. In animation research, Bruderlin and Williams [7], Witkin and Popovic [31], and Gleicher [10] provide interfaces to allow an animator to alter an example motion by modifying key poses within the motion sequence and specifying other constraints on the motion.

This paper adds to previous work by showing how task information can be used directly to adapt examples to new situations. Explicit use of task information should dramatically reduce the number of examples required to represent a behavior, making it feasible to use an example based approach even when the cost of obtaining examples is high.

3 Adapting Grasps to New Object Geometries

One source of variation with which an autonomous character must contend is variation is the environment. One difficult problem is to adapt to variation in the shapes and sizes of objects that must be grasped and manipulated. Consider creating a robot that can repetitively pick up objects and toss them into a bin. This robot is performing a very similar task for each object, but the object geometry may change dramatically from one situation to the next. It would be nice to have a compact description of the grasp or set of grasps that can be used to perform this task.

Pollard [21] [22] presents one method for adapting grasps to new object geometries. This technique assumes that a quality measure can be computed for any grasp. This quality measure is an approximation of the "effort" required to achieve a specified task. It is computed by comparing the space of wrenches (forces and torques) that must be applied to an object for the task with the space of wrenches that can be achieved by a grasp. The space of wrenches that can be achieved by a grasp is obtained by placing a limit on the sum of applied contact forces [13] [9] [16].

To achieve a grasp with the same quality as the example, a character could try to find points on the surface of a new object that allow it to make contact in a way that exactly matches the example grasp. In general this will not be possible, either because there are no suitable corresponding points on the object surface, or because the kinematics of the hand do not allow the hand to reach those points.

If a grasp that exactly matches an example can be found, then the new grasp will be just as good as the example for performing *any* task with the new object. However, this is a much more stringent constraint than is really required. In general, any grasp will be capable of applying to the object a large set of contact wrenches that are not required to accomplish a particular task. This additional grasp capability can be exchanged for more flexibility in placing contacts on the surface of a new object. Details can be found in Pollard [21] [22], and Figure 1 shows an example:

- Figure 1A shows an example grasp, a modeled grasp of a cylinder.
- Seven contacts are extracted from this grasp, as shown in two views in Figure 1B. The task is defined to resist small disturbance wrenches in any direction, and the grasp is generalized to fit this task.
- Good regions corresponding to each of the original seven contacts are then "painted" onto the surface of the new object (Figure 1C). This procedure guarantees that as long as a grasp contains one contact in each of these seven regions, the resulting grasp is guaranteed to be above some quality threshold.
- A search process (described in [22]) found that the coordinate frame of the airplane had to be tilted with respect to that of the example cylinder in order to allow the fingers of the hand to wrap around the wings. The same search process selected the hand configuration shown in Figure 1D.

In summary, this technique allows a single example grasp to be generalized to apply to a wider range of object geometries with no loss in the resulting grasp quality. This approach could allow a compact representation of a space of grasps based on task and general object shape as seen in grasp taxonomies [19] [8], while providing some guidance as to how the examples will need to be adjusted to fit new situations. This general idea of grasp generalization could also be applied to other tasks that involve interaction between a character and

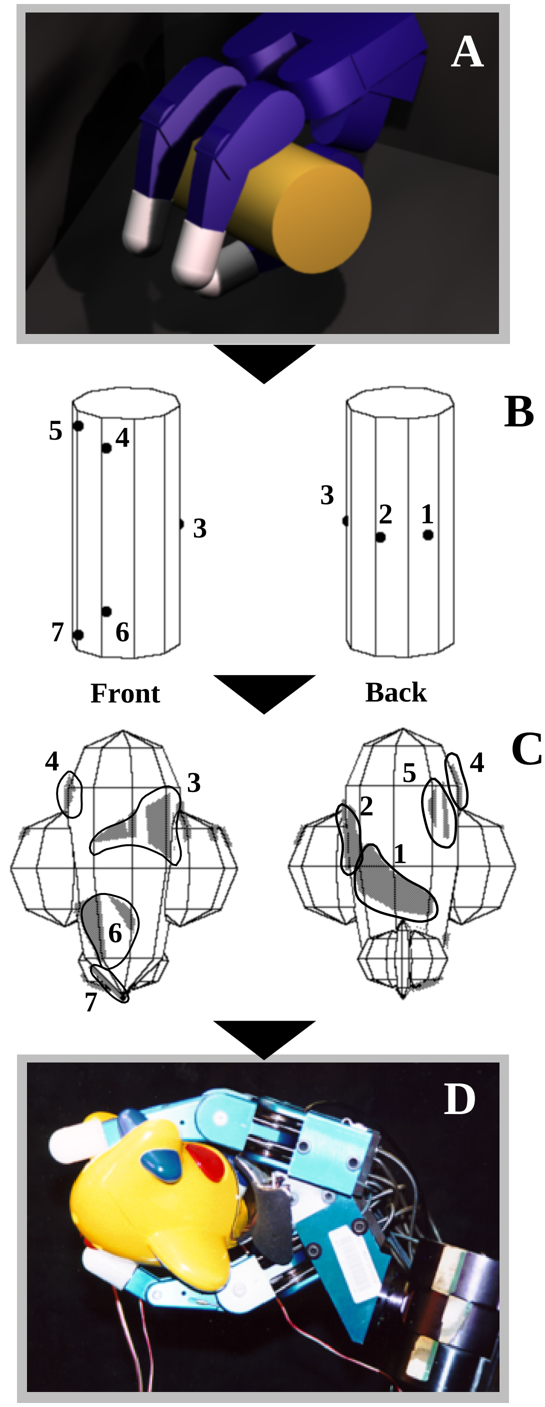

Figure 1: (A) An example grasp of a cylinder; (B) the grasp has seven contacts, shown in two views; (C) good regions corresponding to these contact can be "painted" onto the surface of a new object; (D) a configuration that makes contact in each of the seven regions is guaranteed to be a good grasp for the given task.

its environment. Examples include creating prototype configurations for balancing while rock climbing or for supporting a partner during dance.

4 Adapting Dynamic Behaviors to New Characters

The previous section described a grasp planning algorithm that addressed variation in environment geometry. Another source of variation is in the physical characteristics of a robot or animated character. A well-designed behavior should certainly accommodate minor changes, such as the attachment of an external load to a robot. It should also accommodate larger scale changes. For example, an animator may want to design an entirely new character and have that new character perform in a plausible manner.

If an animator designs a new character from scratch, the likelihood that the character will exactly match an example in a behavior library is small. Characters will tend to have different proportions, and their body parts will have different masses. Because of the large number of parameters required to describe a character's physical characteristics, the size of the input space for this problem is extremely large. In addition, any behavior library is likely to contain only a small number of examples, because the cost of generating each new example is relatively high. It may require measuring the motion of a number of differently sized actors, or it may require the talents of an experienced behavior designer or animator to create and fine-tune the motion for any new character.

To begin to address these problems, Hodgins and Pollard [11] describe a technique that adapts an example behavior to the physical characteristics of a new character. This technique works by scaling control system parameters based on a dynamic analysis of the two characters. The goal is to adjust the control system parameters to achieve motion for the new character that has similar dynamic properties to that of the original.

The animated behaviors used as examples were the running and cycling behaviors described in [12]. The animated motions were computed using dynamic simulation. Each simulation consists of a dynamic model containing the equations of motion for a rigid-body human or humanlike character, constraint equations for the interaction with the ground, parameterized control algorithms for running or bicycling, a graphical representation for viewing the motion, and a user interface for changing the parameters of the simulation. During each simulation time step, the control algorithm computes desired positions and velocities for each joint based on the state of the system and the requirements of the task as specified by the user. Proportional-derivative servos compute joint torques based on the desired and actual value of each joint. The equations of motion of the system are integrated forward in time, taking into account the internal joint torques and the external forces and torques from interactions with the ground plane or other objects. A description of the graphical and dynamic models and an overview of the control algorithms are given in [12].

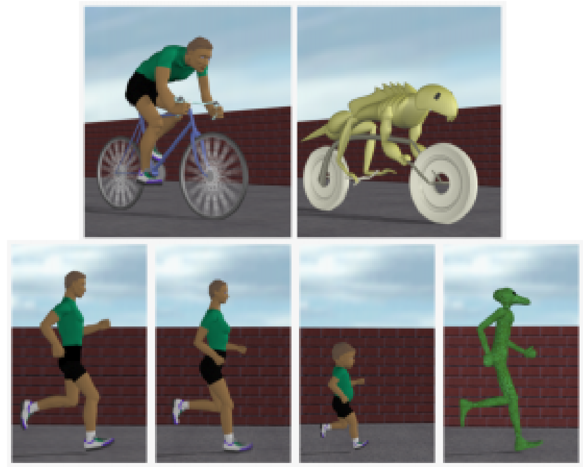

Figure 2: This figure shows results of scaling running and cycling behaviors to new characters. In each case the original behavior was created for the male character on the left and scaled and tuned to adapt it to the other characters shown.

For running and cycling, we had only a single example of each behavior, and we wanted to adapt the behavior to work for a variety of new characters, as shown in Figure 2. If the characters are scale models of each other, then a behavior for the new character can be obtained by scaling control system parameters of the original as described in Raibert and Hodgins [23]. Scale factors for a number of common control system parameters are shown in the *geometric scaling* column of Figure 3. This approach results in dynamically similar motion for the two characters, and it is consistent with models that have been explored for scaling properties in families of animals such as primates or ungulates, which may span up to three orders of magnitude in size [18].

In general, however, individuals will not be scale models of one another, and a geometric scaling approximation will not be adequate for generating motion for a new character. There are two problems with this approximation:

- No single scale factor is adequate. For example, a scale factor computed based on relative leg lengths may not be appropriate for scaling the gains that control arm motion.
- Dynamically similar behavior cannot in general be achieved between two arbitrary models.

Quantity	Units	Geom. Scaling	Mass Scaling
Basic variables			
length	L	L	–
time	T	$L^{1/2}$	–
force	F	L^3	M
torque	FL	L^4	IL^{-1}
Motion variables			
displacement	L	L	–
velocity	LT^{-1}	$L^{1/2}$	–
acceleration	LT^{-2}	1	–
ang displacement	–	1	–
ang velocity	T^{-1}	$L^{-1/2}$	–
ang acceleration	T^{-2}	L^{-1}	–
Mechanical parms.			
mass	$FL^{-1}T^2$	L^3	M
stiffness	FL^{-1}	L^2	ML^{-1}
damping	$FL^{-1}T$	$L^{5/2}$	$ML^{-1/2}$
moment of inertia	FLT^2	L^5	I
torsional stiffness	FL	L^4	IL^{-1}
torsional damping	FLT	$L^{9/2}$	$IL^{-1/2}$

Figure 3: Scaling rules that capture differences in size, mass, and moment of inertia. The geometric scaling factor is derived assuming uniform scaling by factor L in all dimensions (geometric similarity), and assuming that the acceleration of gravity and the density of the material are invariant to scale. The mass scaling factor assumes also that mass scales by factor M and moment of inertia scales by factor I. A "–" in the mass scaling column indicates that the mass scaling rule is the same as the geometric scaling rule.

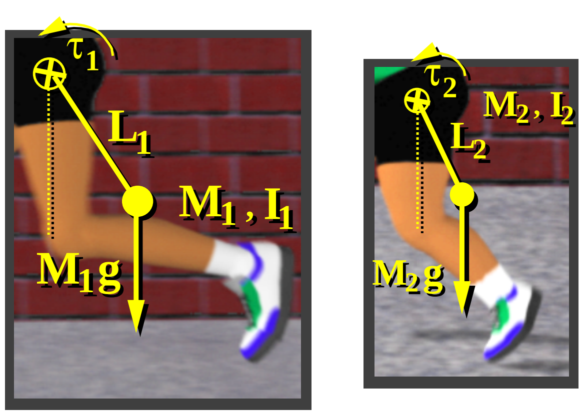

Figure 4: The leg of the child (right) is not a scale model of the woman's leg. A scale factor can be computed for gains at the hip joint during flight by assuming that the dimensionless acceleration of the leg produced by torque at the hip is constant for the two systems.

The first point is addressed by computing different scale factors for different parts of the system. The scale factors for gains at a particular joint, for example, are computed based on the function of that joint. Gains at the hip are primarily responsible for swinging the leg forward during flight, and so the gains for the hip during flight will be scaled based on the physical characteristics of the leg (Figure 4).

The second point is also illustrated with Figure 4. Measured properties of the leg are mass (M), length (L), and inertia (I). Dynamic similarity requires that the dimensionless group I/ML^2 be constant for the two systems, but this is a parameter over which we have no control, and so achieving dynamically similar motion may not be possible. An approximation is to assume constant dimensionless acceleration for the two systems. This assumption, along with the requirement that we need coordinated motion (e.g. the arms cannot be swung with a frequency different from the legs) results in the scaling factors shown in the *mass scaling* column of Figure 3. Further details can be found in [11].

This technique has proven to be a good first step to scaling behaviors to new characters, but it is only an approximate scaling technique. The mass scaling approach is an approximation because it sets up the goal of achieving similar looking motion (in terms of joint accelerations) at different phases of the behavior. Equivalent motion in this sense will not result in exactly the same behavior unless the characters do happen to be scale models of each other. As a simple example, a person wearing a heavy backpack will have a tendency to lean forward while walking or running in order to maintain balance. This type of adjustment is not accounted for if the only goal is to achieve constant dimensionless acceleration.

In the end, this simple scaling approach worked well because a variety of task information is built into the running and cycling controllers to help counteract errors. In the case of the dynamic behaviors shown here, task information includes a state machine, feedback controllers, and some inverse kinematics (e.g. for foot placement during landing). A search over a small number of high-level control parameters was also an important step in the scaling process, and was used to tune the motion for the new characters shown in Figure 2.

5 Discussion

This paper suggests that examples can be adapted directly to new situations using an analysis based on task requirements. Two applications were shown to support this argument. The first used a static analysis based on forces and torques to adapt a grasp to a new object geometry. The second application used a dynamic analysis based on accelerations to adapt running and cycling behaviors to new animated characters.

Perhaps force and acceleration analyses can be combined to develop better scaling laws for dynamic tasks involving interaction with the environment. There is some evidence that people make use of such scaling laws to achieve different goals. For example, Schaal, Sternad and Atkeson [25] suggest that a dynamic scaling relationship can be observed in the paddle motion of subjects asked to paddle-juggle a ball at three different heights.

A similar approach could be used to animate tasks such as dribbling, catching, and throwing a basketball. These behaviors could be represented compactly with a small number of examples, and the examples could be dynamically scaled to allow the character to dribble at different heights, catch balls coming in at different velocities, and throw the ball along different trajectories. Here, the characteristics of the interaction between the hands and the ball could be adapted to appropriately scale the ball's behavior.

To speculate further, we can look at a motion such as running in a similar way. In this case, the character itself is the manipulated object. In other words, the interesting interaction occurs between the character's foot and the ground. Perhaps a similar approach can be used to more accurately scale the landing and liftoff parts of the running motion by controlling the impedance of the entire body as seen from the foot/ground contact.

What is the advantage of having good algorithms to adapt behaviors to variations in environment geometry, physical characteristics, or tasks? Such algorithms should at least reduce the number of examples required to represent a behavior. In grasping, examples could be used to cover the qualitatively different families of grasps that might be identified in a grasp taxonomy [19] [8], and a force/torque analysis could be used to ensure that grasps can be adapted to changes in geometry for objects within the same family. For dynamic motions such as manipulating a basketball, perhaps the extra "storage space" for examples could be used to cover a range of styles–something that currently seems difficult to capture quantitatively.

Acknowledgments

This project was supported in part by NSF Grant No. CDA-9704694, by NSF NYI Grant No. IRI-9457621, by Mitsubishi Electric Research Laboratory, and by a Packard Fellowship.

References

[1] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning for control. *Artificial Intelligence Review*, (in press).

[2] C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *International Conference on Machine Learning*, 1997.

[3] N. A. Bernstein. On dexterity and its development. In M. L. Latash and M. T. Turvey, editors, *Dexterity and its Development*. Lawrence Erlbaum Associates, New Jersey, 1996.

[4] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 47–54. ACM SIGGRAPH, ACM Press, July 1995.

[5] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[6] A. Bruderlin and T. W. Calvert. Goal-directed, dynamic animation of human walking. In *Computer Graphics (SIGGRAPH 89 Proceedings)*, volume 23, pages 233–242, July 1989.

[7] A. Bruderlin and L. Williams. Motion signal processing. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 97–104. ACM SIGGRAPH, Addison Wesley, August 1995.

[8] M. R. Cutkosky and R. D. Howe. Human grasp choice and robot grasp analysis. In S.T. Venkataraman and T. Iberall, editors, *Dextrous Robot Hands*. Springer-Verlag, New York, 1990.

[9] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. IEEE ICRA*, Nice, France, May 1992.

[10] M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 139–148, Providence, RI, April 1997.

[11] J. K. Hodgins and N. S. Pollard. Adapting simulated behaviors for new characters. In *SIGGRAPH 97 Proceedings*, pages 153–162. ACM SIGGRAPH, Addison Wesley, 1997.

[12] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 71–78. ACM SIGGRAPH, Addison Wesley, August 1995.

[13] D. G. Kirkpatrick, B. Mishra, and C. K. Yap. Quantitative steinitz's theorems with applications to multifingered grasping. In *Proc. 20th ACM Symposium on Theory of Computing*, Baltimore, Maryland, May 1990.

[14] Y. Koga, K. Kondo, J. Kuffner, and J. C. Latombe. Planning motions with intentions. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 395–408. ACM SIGGRAPH, ACM Press, July 1994.

[15] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[16] Z. Li and S. Sastry. Optimal grasping by multifingered robot hands. In *Proc. IEEE Intl. Conference on Robotics and Automation*, Raleigh, North Carolina, 1987.

[17] Z. Liu, S. J. Gortler, and M. F. Cohen. Hierarchical spacetime control. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994.

[18] T. A. McMahon. *Muscles, Reflexes, and Locomotion*. Princeton University Press, Princeton, 1984.

[19] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38B(4), 1956.

[20] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *SIGGRAPH 96 Proceedings*, Annual Conference Series, pages 205–216. ACM SIGGRAPH, ACM Press, July 1996.

[21] N. S. Pollard. Synthesizing grasps from generalized prototypes. In *Proc. IEEE Intl. Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996.

[22] N. S. Pollard. Parallel algorithms for synthesis of whole-hand grasps. In *Proc. IEEE Intl. Conference on Robotics and Automation*, Albuquerque, NM, 1997.

[23] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Proceedings)*, volume 25, pages 349–358, July 1991.

[24] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics (SIGGRAPH 87 Proceedings)*, volume 21, pages 25–34, July 1987.

[25] S. Schaal, D. Sternad, and C. G. Atkeson. One-handed juggling: A dynamical approach to a rhythmic movement task. *Journal of Motor Behavior*, 28(2):165–183, 1996.

[26] C. Stanfill and D. Waltz. Towards memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[27] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 43–50. ACM SIGGRAPH, ACM Press, July 1994.

[28] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 91–96. ACM SIGGRAPH, Addison Wesley, August 1995.

[29] D. J. Wiley and J. K. Hahn. Interpolation synthesis for articulated figure motion. In *VRAIS '97*, pages 156–160, Albuquerque, New Mexico, 1997.

[30] A. Witkin and M. Kass. Spacetime constraints. In J. Dill, editor, *Computer Graphics (SIGGRAPH 88 Proceedings)*, volume 22, pages 159–168, August 1988.

[31] A. Witkin and Z. Popović. Motion warping. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 105–108. ACM SIGGRAPH, Addison Wesley, August 1995.

[32] W. L. Wooten. *Simulation of Leaping, Tumbling, Landing, and Balancing Humans*. PhD thesis, Georgia Institute of Technology, Department of Computer Science, 1998.

[33] X. Zhao, D. Tolani, B. Ting, and N. I. Badler. Simulating human movements using optimal control. In *Eurographics Workshop on Computer Animation and Simulation '96*, pages 109–120, 1996.