

# Animating Athletic Motion Planning By Example

Ronald A. Metoyer

Jessica K. Hodgins

College of Computing and Graphics, Visualization, and Usability Center  
Georgia Institute of Technology  
801 Atlantic Drive  
Atlanta, GA 30332-0280  
e-mail: [metoyer|jkh]@cc.gatech.edu

## Abstract

Character animation is usually reserved for highly skilled animators and computer programmers because few of the available tools allow the novice or casual user to create compelling animated content. In this paper, we explore a partial solution to this problem which lets the user coach animated characters by sketching their trajectories on the ground plane. The details of the motion are then computed with simulation. We create memory-based control functions for the high-level behaviors from examples supplied by the user and from real-world data of the behavior. The control function for the desired behavior is implemented through a lookup table using a  $K$ -nearest neighbor approximation algorithm. To demonstrate this approach, we present a system for defining the behaviors of defensive characters playing American football. The characters are implemented using either point-masses or dynamically simulated biped robots. We evaluate the quality of the coached behaviors by comparing the resulting trajectories to data from human players. We also assess the influence of the user's coaching examples by demonstrating that a user can construct a particular style of play.

*Key words:* animation, behavioral control, physical simulation, machine learning

## 1 Introduction

No matter how realistic a character may look, if it behaves in an unnatural way, the illusion of reality is lost. Animated shorts, video games, interactive virtual environments, and training simulation environments all require animated characters whose high-level behavior conveys an appearance of intelligence. For example, the opponent agents in Quake are most appealing when they appear as proficient in combat as a skilled human opponent. In a sports training environment, unrealistic behaviors are unacceptable because they can lead to negative training.

High-level behaviors govern the way a character moves within the environment to achieve its goals in the presence of obstacles and other characters. Two-dimensional navigation control is necessary in any three-dimensional

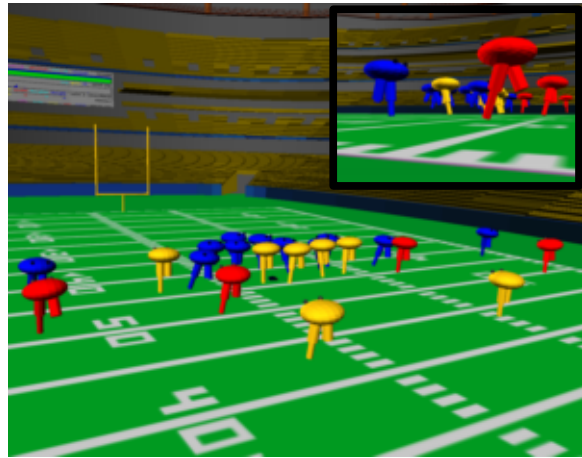


Figure 1: Two teams of dynamically simulated bipeds playing American football. The offensive players (blue) are tracking pre-defined offensive routes from real football data. Seven defensive players (yellow) are also tracking pre-defined routes while the four red defensive players are controlled by the behavior function table. Collisions are enforced between all players.

scene with locomoting characters. For example, path planning algorithms allow a character in a video game to avoid collisions with other moving characters and static obstacles. High-level behaviors can also tell a character how to move to accomplish a task such as defending a soccer goal.

We need intuitive interfaces for creating compelling and realistic high-level behaviors because the people who would like to create the content are not always experts in animation, control, or computer programming. For example, a quarterback training simulator should allow a football coach to set up a scenario that is appropriate for a particular trainee. A sports video game could allow the user to demonstrate defensive maneuvers and customize the playing style of his team. Similarly, a child might want to create an animation by directing a swarm of bugs to navigate a particular terrain. Architectural visualizations would benefit from a two-dimensional interface for

creating animated figures that navigate through a particular three-dimensional architectural structure.

In this paper, we explore a data-driven approach to defining navigation control where much of the data is supplied directly by the user via an intuitive mouse interface. We use a real-time, memory-based technique that builds local approximations of the appropriate action based on data stored in a table. The data is obtained from coaching examples provided by the user and from observations of real-world scenes[11]. We use point-mass simulations as well as three-dimensional dynamically simulated characters that possess low-level locomotion primitives (Figure 1) to execute the output of the navigation control. We have used this system to implement defensive behaviors for a man-to-man defense and a limited zone defense in American football. We demonstrate the effectiveness of coaching examples by comparing the performance of the coached behaviors to recorded human behavior. We also show that the user can provide coaching examples to create particular styles of defensive play.

## 2 Background

We build on a considerable history of work in memory-based techniques for creating both high and low-level controllers. In the machine learning community, memory-based learning has been used successfully in robot control. Atkeson and colleagues survey the use of locally weighted learning for robot control tasks[2]. Moore investigates efficient memory-based techniques for robot control[13]. Aha and Salzberg explore the use of nearest-neighbor algorithms for a robot that learns to catch a ball[1]. Researchers have also explored the use of learning from examples to develop road-following controllers for vehicles. The ALVINN system uses a neural network to train off-line on examples of road-following[14]. The ELVIS system learns the eigenvectors of an input image and steering commands for road following examples and projects new examples into this eigenvector feature space to determine steering commands for new situations[8]. In computer graphics, Grzeszczuk, Terzopoulos, and Hinton used neural networks to learn the dynamics and control of several dynamic systems including a rocket ship, a car, and a dolphin[7].

Many other researchers in computer graphics have explored data-driven approaches that combine multiple motion sequences to produce new motion. Unuma, Anjyo, and Takeuchi use Fourier interpolation to generate walking and running gaits that express a variety of emotions[16]. Wiley and Hahn use a tri-linear interpolation pyramid and time warping to create generalized pointing as well as walking on sloped terrain[17]. Rose, Cohen, and Bodenheimer use a radial basis function model to generalize behaviors such as walking and running for speed and angle of the terrain[15]. Their work also emphasizes parametric control of emotional expressiveness for a set of base behaviors.

The goal of our research is to explore whether intu-

itive interfaces can be designed that allow the novice to construct animations effectively. Recently, superb work has been done on a related problem, intuitive interfaces that allow the novice to build three-dimensional graphical models. These systems used a pen and/or mouse to provide the user with intuitive control within a constrained domain. Both Egli and Zeleznik and their colleagues explored free-hand sketching techniques for creating three-dimensional rectilinear models[5, 18] while Igarashi, Matsuoka, and Tanaka provided an intuitive interface for modeling rounded freeform objects[10]. Our interface builds on these ideas by using a mouse to provide the high-level control examples in the two-dimensional plane and a dynamic simulation as a constraint to produce the details of the low-level animated motion.

Several researchers have explored intuitive interfaces for animation or character direction in virtual environments. Blumberg and Galyean allow the user to direct a character at the motivational, task, and motor levels[4]. Johnson and his colleagues embedded an instrumented skeleton within a plush toy and allowed users to manipulate the toy. Their system recognized common gestures made by the user and interpreted them as commands for the control of an interactive animated character[12]. Each of these techniques constrains the number of degrees of freedom that the user must control to make the problem tractable.

## 3 American Football

In this paper, we explore a mouse-based interface for coaching reactive behaviors for the defensive-back positions in American football. We first develop a behavior function table based on an existing database of tracked football plays[11]. We then allow the user to modify the behavior by adding coaching examples of the desired behavior. We also build behaviors starting from an empty table for other tests. We present the behavior results with both an offensive/defensive pair of point-masses and two full teams of biped robots. The offensive players and some of the defensive players run pre-defined routes from the database of football plays while the actions of selected defensive players are determined by the behavior function table.

We focus on two types of defense in football: man-to-man and zone. The characteristics of each defensive strategy determine the features that are used to index into the behavior function table as well as the actions that result from the function query. The man-to-man strategy requires that the defender focus on a single offensive receiver and maintain a position near the receiver as he moves down the field. The defender must stay close enough to the receiver to prevent the completion of a short pass but allow enough space to prevent the receiver from getting behind him to complete a long pass[3, 6]. The zone strategy requires that the defender focus on defending any pass within a given area of the field rather than focusing on a specific player. When the play begins, the defender observes the offensive players and takes the

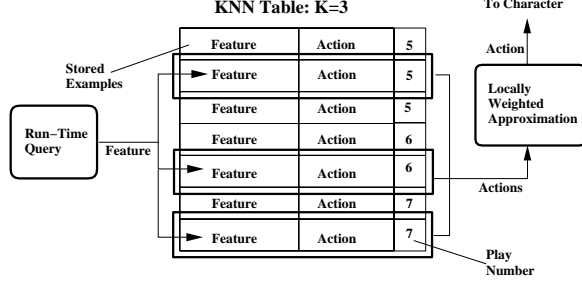


Figure 2: During training, the algorithm stores all data. At query time, it finds the  $K$ -nearest neighbor examples in the table and uses them to compute a local approximation to the appropriate action.

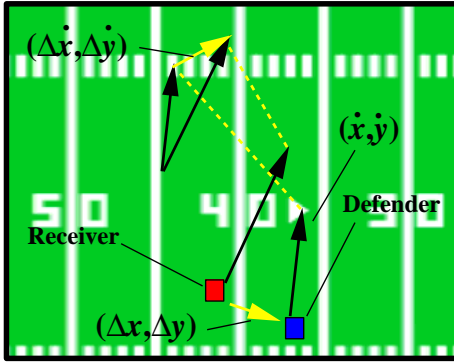


Figure 3: The man-to-man defensive strategy depends on the absolute velocity of the defender  $(\dot{x}, \dot{y})$ , his position with respect to the receiver  $(\Delta x, \Delta y)$ , and his velocity with respect to the receiver  $(\Delta \dot{x}, \Delta \dot{y})$ . The corresponding action is the desired position of the defender with respect to the receiver  $(\Delta x_d, \Delta y_d)$ .

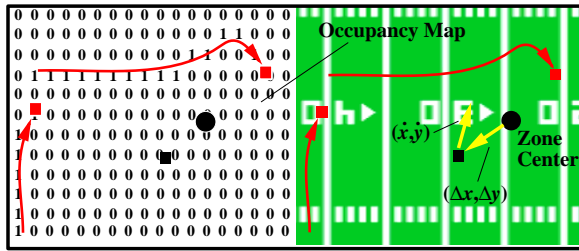


Figure 4: The zone defensive strategy depends on the absolute velocity of the defender  $(\dot{x}, \dot{y})$ , the relative position of the defender with respect to the zone center  $(\Delta x, \Delta y)$ , and a measure of the offensive activity in the assigned zone area (occupancy map). The corresponding action is the desired position of the defender with respect to the zone center  $(\Delta x_d, \Delta y_d)$ .

appropriate position within his assigned zone based on the movement of the offensive players[3, 6].

#### 4 Behavior Representation

Memory-based learning is a lazy learning technique because all behavior examples are stored in a table and

actual learning or function approximation does not take place until a query has been made. The query is a feature vector that represents the current state or situation. When a query is made, the  $K$  feature vectors that are closest to the query are retrieved from the table along with their accompanying actions. These  $K$  feature/action pairs are used to build a distance-weighted local approximation of the action (Figure 2).

Figure 3 illustrates the three features used in the man-to-man defensive strategy: the defensive player's absolute velocity  $(\dot{x}, \dot{y})$ , his position relative to the receiver  $(\Delta x, \Delta y)$ , and his velocity relative to the receiver  $(\Delta \dot{x}, \Delta \dot{y})$ . The action is the desired relative position of the defender with respect to the receiver:  $(\Delta x_d, \Delta y_d)$ .

Figure 4 shows the three features used in the zone defensive strategy: the defender's absolute velocity  $(\dot{x}, \dot{y})$ , his position relative to the zone center  $(\Delta x, \Delta y)$ , and the general offensive activity in his zone area. The general activity within a zone area is determined with a low resolution occupancy map (16x24 m) of the offensive players within the general zone area. If an offensive player occupies a discretized position on the field over the course of a play, a value of one is assigned to that position, creating a map of offensive player travel within the zone area. The maps are compared in a bitwise fashion and the distance between two maps is the number of entries that differ. The corresponding action is the desired position of the defender with respect to the zone center:  $(\Delta x_d, \Delta y_d)$ .

A feature vector  $p$  is represented as  $(a_1(p), a_2(p), \dots, a_n(p))$  where  $a_r(p)$  denotes the  $r$ th attribute of the feature vector  $p$ . The distance between two feature vectors  $p_i$  and  $p_j$  is

$$d(p_i, p_j) = \sqrt{\sum_{r=1}^n (a_r(p_i) - a_r(p_j))^2}$$

The weighting for a particular neighbor  $i$  given a query,  $q$ , is  $w_i = 1/d(p_q, p_i)^2$ . The distance-weighted action is then

$$f(p_q) = \frac{\sum_{i=1}^K w_i f(p_i)}{\sum_{i=1}^K w_i}$$

where  $K$  is the number of nearest neighbors used in the approximation and  $f(p_i)$  is the action associated with the instance  $p_i$ . This locally approximated action is then used as the output.

#### 5 Sources of Data

The  $K$ -nearest neighbor table is built on examples of the desired behavior. These examples can come from measurements of the real world or from data sketched by the user. In Section 7, we describe experiments that use these sources of data both independently and together. A behavior example is defined as a player's defensive trajectory over time. Regardless of the source, the behavior example is sampled at 30Hz and these samples define the



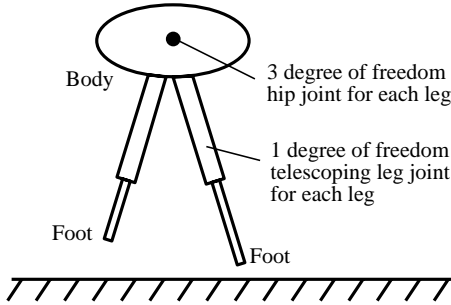


Figure 7: The biped robot consists of a body and two telescoping legs. Each leg has three degrees of freedom at the hip and a fourth degree of freedom for the length of the leg.

## 6 Simulations for Low-Level Behaviors

We demonstrate the performance of the behavior function table on both point-mass simulations and dynamically simulated biped robots. The point-masses have a mass of 100kg and a maximum velocity of 10m/s. A spring and damper connect the point-mass to the desired position on the field. The spring and damper position gain,  $k_p$ , was 7000 and the velocity damping gain,  $k_v$ , was 700 in our experiments.

The dynamically simulated biped consists of 5 body parts and 8 degrees of freedom (Figure 7). The biped has a leg length of 0.9 meters (approximately the length of the leg of a person 2 meters tall) and uses control laws described in [9]. Because the biped robots are not as agile as human football players, the features are scaled before lookup in the behavior function table and the inverse of the scaling factor is applied to the resulting action. The tracked humans reach maximum velocities of 10m/s whereas the biped robot has a maximum velocity of approximately 4m/s resulting in a scaling factor of 2.5.

## 7 Results

We ran several tests to evaluate the performance of the system. The first test illustrates the improvement provided by combining coaching with a baseline behavior function table built from the database of tracked football plays. We extracted 30 examples of man-to-man defensive behavior as played by the outside linebacker and cornerback positions. The corresponding feature/action pairs for 29 of these examples were entered into the table to serve as the baseline for man-to-man defensive behavior. Although this table performs reasonably well on many plays, in some cases, the performance is not particularly good.

When the user provides partial or full examples of the desired behavior for several of the 29 plays, the performance improves for the test play that was withheld from the table (Figure 8). The user provided 8 example sequences that were the equivalent of approximately 2 full plays of 3s each. Of the actions used in the final trajectory, 30% were from coached data while the remaining

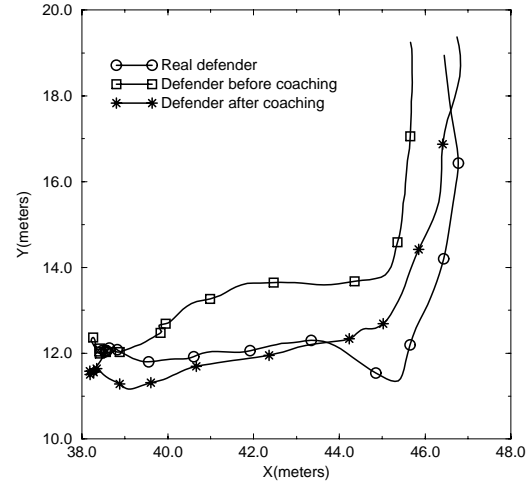


Figure 8: The line marked with squares shows the performance of the point-mass using a behavior built from the database of football plays. The line marked with stars shows the performance after the table has been augmented with coaching examples while the line marked with circles shows the response of the digitized actual defender. The symbols represent time stamps as the play moves down the field from left to right.

70% were from the original table data.

Figure 9 shows the results of coaching a character to play a generic man-to-man defense. Starting from an empty table, the character was coached on a set of randomly chosen plays from the database of real football plays. The graph depicts the defender's performance for a play that was not included in the coaching drills. The performance of the coached player is similar to that of the real player not just in the path taken but also in the timing of the play.

To be useful as a tool, the system must allow the user to mold the behavior of the character. We ran two tests to exercise this aspect of the system performance. In each test, the user was to create two distinctly different styles of man-to-man defense. Figure 10 shows a defender coached to have an inside defensive bias and another coached to have an outside defensive bias. This graph is for a play that was not in the coaching set. A defender might take such a bias if a defensive strategy provides him with support from other defensive players on either side. We ran a similar experiment to create both a tight and a loose man-to-man defensive style. The need for these two styles might arise in a football training environment where the content creator would want a defender who plays a tight man-to-man defense for a "goal line" scenario but a loose man-to-man for a scenario of "third down and long."

For this experiment, we ran user tests to determine if subjects other than the authors could use the coaching interface to create the two styles of defense. The subjects were given approximately five minutes to fa-

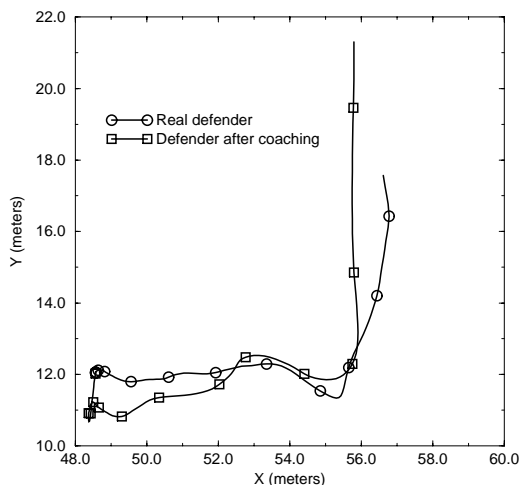


Figure 9: A point-mass character coached to play generic man-to-man competitive defense. Performance against a play from the database is similar to that of the real defender that was recorded for this particular play. The play moves down the field from left to right.

miliarize themselves with the dynamics of the coaching interface. They were then presented with two separate movies of biped robots playing a man-to-man defense against biped receivers. One video represented a tight man-to-man defense while the other represented a loose man-to-man defense. The subjects were asked to coach a point-mass character to play a loose man-to-man defense against a point-mass receiver for six plays. After training on these six plays, the resultant behavior table was applied to an unseen and uncoached play to determine if the subject was satisfied with the resulting loose man-to-man defensive behavior. The process was repeated for the tight man-to-man defense.

On a scale of 1-5, all of the subjects ranked their ability to control the point-mass a 3 or above, and rated their satisfaction with the resulting behavior on the unseen play a 3 or above. Figure 11 shows a graph of the point-mass performance on an unseen play after being coached by one of our subjects. The loose variation was coached for a total of 1015 feature/action pairs, the equivalent of approximately 11 plays of 3s each. The tight variation was coached for a total of 1332 feature/action pairs, the equivalent of approximately 15 plays of 3s each.

Figure 12 shows a graph of the average separation distance between the defender and the receiver for both the loose and tight version of the man-to-man defense for each test subject as well as for the demonstrated man-to-man play that they were attempting to imitate. Subjects 2-6 clearly demonstrate the two distinct styles of man-to-man defense, but subjects 7 and 8 were unable to produce the two styles of defense. Both subjects 7 and 8 are left-handed, but use their right hands for typical mouse manipulations. All other subjects are right-handed.

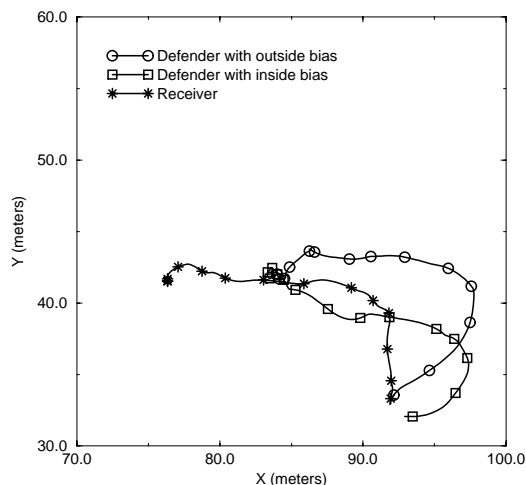


Figure 10: The line marked with circles represents a point-mass defender coached to play a man-to-man defense with an outside position bias. The line marked with squares represents a defender coached to play with a bias to the inside of the receiver. Both behaviors were coached starting from an empty table. The play moves down the field from left to right. The sideline is at the top of the graph, and the player with the outside bias defends closest to the sideline.

Among the right-handed subjects, those with better control over the character were clearly able to produce more consistent examples and this consistency was reflected in the resulting behavior. The subject's level of football knowledge appeared to have little effect as long as the subjects were consistent in their examples.

In the next example, we show that the user can train the character for a limited zone defense. The defender was drilled on three types of plays, with multiple variations of each type. There are three basic zone defensive rules for a linebacker [3]. First, if the halfback to the linebacker's side runs a wide route, the defender must stay wide to protect against the pass to the halfback. Second, if the halfback runs a stop route, the wide receiver will be running a route to the outside. The defender must move to a wide position in the zone to protect against the wide receiver outside route. And finally, if the halfback runs an angle route, the wide receiver is running a deep-inside route. The defender must get to a deep position in the zone and fade to the inside to protect against the wide receiver inside route.

Figure 13 shows an example of the second rule. The offensive player routes were synthetically created for each of the zone examples because the database of football plays did not contain a sufficient number of examples. The defender reacts correctly for a play that was not used in the coaching drills. He moves to a wide position to protect against the wide receiver out route as stated in the second rule.

All of the preceding examples used a point-mass sim-



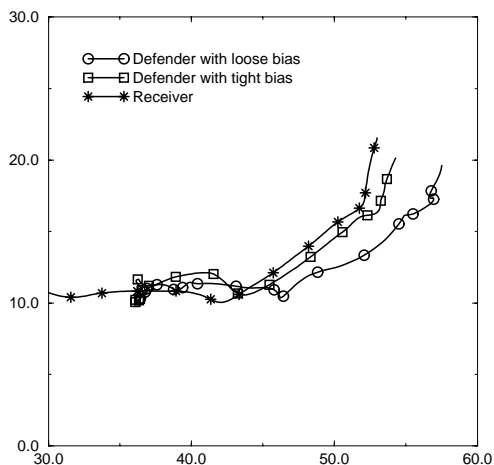


Figure 11: The line marked with circles represents a point-mass defender who was coached to play a loose man-to-man. The line marked with squares represents a defender who was coached to play a tight man-to-man. Both behaviors were coached starting from an empty table. The play moves down the field from left to right.

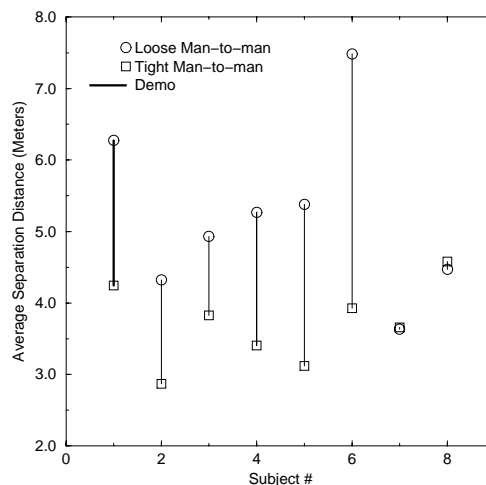


Figure 12: Average separation distance between defender and receiver for user test subjects coaching a loose man-to-man and tight man-to-man defense. The bold line represents the average separation distance for the demonstration play.

ulation to compute the low-level motion but we can also use simulated bipeds as the football players. Figure 14 shows the performance of a simulated biped robot defender. The robot is controlled with a behavior function table that did not include this particular play. The biped trajectory is compared to that of the human player where the human data has been scaled by  $1/2.5$  to match the abilities of the biped.

In each of these examples, the offensive characters are not responsive to the defense and are running predefined routes from the database of football plays or from hand-drawn routes in the case of the zone. This simplification is often reasonable for receivers because their routes must be followed closely after a play has started. The quarterback will often throw the ball to a predetermined spot on the field before the receiver arrives or even looks toward the quarterback.

## 8 Discussion

This paper illustrates how coaching can be used to create or refine high-level behaviors for animated characters and demonstrates the approach in American football. We built a man-to-man defense behavior based on an initial database of examples and applied it to both a point-mass and a biped robot. We demonstrated the power of the user interface by using coaching to modify this baseline behavior function table as well as to define man-to-man and zone defense behaviors from an initially empty table. We performed user tests and showed that subjects could produce both a tight and a loose man-to-man defense.

The selection of appropriate features is an important aspect of our approach because it determines the dimen-

sionality of the table and therefore the amount of data necessary to populate the table. Unnecessary features result in wasted storage space while too few features can result in an inability to model the desired function. We selected a small set of appropriate features based on our domain knowledge. Given sufficient data, techniques such as principal components analysis might compute the appropriate features automatically. Scaling of the features also affects the performance of our system. We do not normalize the features for the man-to-man or for the zone examples because they were on the same order of magnitude. We did, however, weight the relative position by a factor of 3.0 to increase its importance. For the zone example, our occupancy map feature is potentially subject to noise problems because offensive situations that are shifted with respect to the zone center may not appear similar. This problem could be addressed by using pattern matching techniques that are invariant with respect to shift.

Although we have used only real-world and coaching examples to control the characters in this paper, we believe that the next step towards easily programmed and controlled characters is a combination of several techniques including coaching examples, real-world data, hand-programmed reactive behaviors, and finite state machines for switching between behaviors. Combining coaching with other techniques would allow the animator to influence behavior when desired but would free the animator from providing examples for every situation.

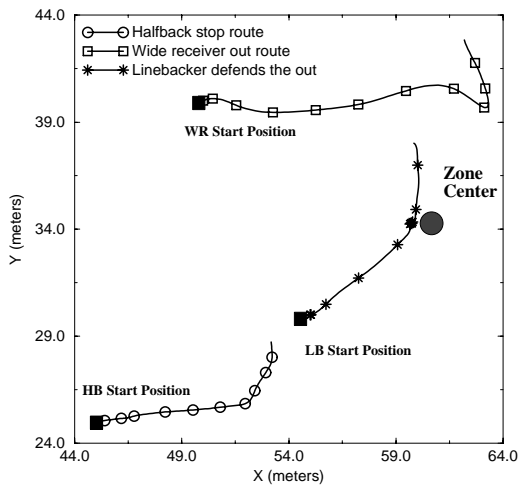


Figure 13: After coaching on several zone play examples, the character effectively defends the zone for this play where the halfback runs a stop pattern. The linebacker retreats to his zone area and correctly moves to the outside to guard against the wide receiver out route.

### Acknowledgements

We thank Eugene Zhange for the biped collision detection and response implementation. This project was supported in part by a Lucent Technologies Fellowship, NSF NYI Grant No. IRI-9457621, and a Packard Fellowship.

### References

- [1] D.W. Aha and S. L. Salzberg. Learning to catch: Applying nearest neighbor algorithms to dynamic control tasks. In *Fourth International Workshop on Artificial Intelligence and Statistics*, pages 363–368, 1993.
- [2] C.G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning for control. *Artificial Intelligence Review*, pages 75–113, 1996.
- [3] T. Bass. *Play Football the NFL Way*. St. Martin's, New York, NY, 1992.
- [4] B.M. Blumberg and T.A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. *Proceedings of SIGGRAPH 95*, pages 47–54, August 1995.
- [5] L. Eggli, C. Hsu, G. Elber, and B. Bruderlin. Inferring 3D models from freehand sketches and constraints. *Computer Aided Design*, 29(2):101–112, 1997.
- [6] T. Flores and B. O'Conner. *Coaching Football*. Masters Press, Indianapolis, IN, 1993.
- [7] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. *Proceedings of SIGGRAPH 98*, pages 9–20, July 1998.
- [8] J. Hancock and C. Thorpe. ELVIS: Eigenvectors for land vehicle image system. In *Proceedings of the International*

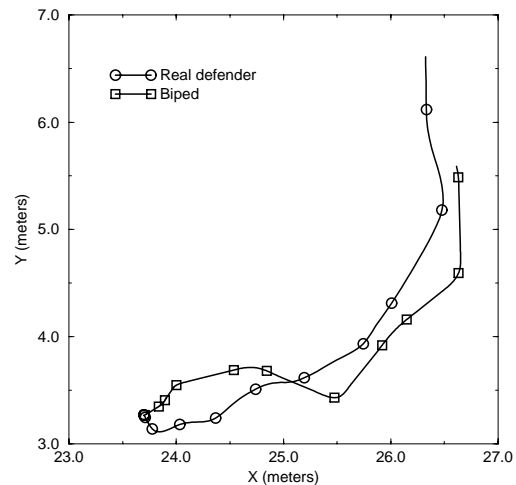


Figure 14: The biped defensive trajectory compared to the recorded human defensive trajectory.

*Conference on Intelligent Robots and Systems (IROS 95)*, pages 35–40, 1995.

- [9] J. K. Hodgins, J. Koehling, and M. H. Raibert. Running experiments with a planar biped. In *Robotics Research: The Third International Symposium*, 1986.
- [10] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3D freeform design. *Proceedings of SIGGRAPH 99*, pages 409–416, August 1999.
- [11] S. Intille and A.F. Bobick. MIT Media Lab football trajectory database. <http://vismod.www.media.mit.edu/vismod/demos/football>, 1998.
- [12] M.P. Johnson, A. Wilson, B. Blumberg, C. Kline, and A. Bobick. Sympathetic interfaces: Using a plush toy to direct synthetic characters. In *Proceedings of CHI '99*, pages 152–158, 1999.
- [13] A.W. Moore. *Efficient Memory-based Learning for Robot Control*. PhD thesis, University of Cambridge, 1990.
- [14] D. Pomerleau. *Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving*. Kluwer Academic Publishing, 1993.
- [15] C. Rose, M. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
- [16] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. *Proceedings of SIGGRAPH 95*, pages 91–96, August 1995.
- [17] D. J. Wiley and J. K. Hahn. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 156–160, March 1997.
- [18] R.C. Zeleznik, K.P. Herndon, and J.F. Hughes. Sketch: An interface for sketching 3D scenes. *Proceedings of SIGGRAPH 96*, pages 163–170, August 1996.