

Anticipation from Example

Victor Zordan^{†*}, Adriano Macchietto[†], Jose Medina[†], Marc Soriano[†], Chun-Chih Wu[†], Ronald Metoyer[‡], Robert Rose[‡]
[†]University of California, Riverside [‡] Oregon State University

Abstract

Automatically generated anticipation is a largely overlooked component of response in character motion for computer animation. We present an approach for generating anticipation to unexpected interactions with examples taken from human motion capture data. Our system generates animation by quickly selecting an anticipatory action using a Support Vector Machine (SVM) which is trained offline to distinguish the characteristics of a given scenario according to a metric that assesses predicted damage and energy expenditure for the character. We show our results for a character that can anticipate by blocking or dodging a threat coming from a variety of locations and targeting any part of the body, from head to toe.

Keywords: motion capture, human animation, behavior control

1 Introduction

Anticipation behavior has been largely overlooked in computer-generated characters, especially in interactive settings such as virtual environments and electronic games where such motions must be computed automatically. While responding *after* an interaction has received considerable attention and is necessary to uphold the physical realism of contact resulting from an interaction, anticipatory response *before* an interaction is an important component for making characters appear alert to their environment and concerned about their own well-being. In this paper, we introduce a novel technique for generating anticipation that selects from a database of possible motion capture examples of anticipation based on the specific conditions of an impending interaction. To accomplish this task, we employ a supervised learning method which is trained on the best anticipation to use for a given scenario computed by weighing the potential damage sustained relative to the amount of effort required to carry out a particular anticipatory act.

As our testbed we focus on making a character anticipate and block a threat coming from a range of directions, heights, and speeds. We focus our domain on a character which starts from an idle, standing state (rather than anticipating interactions starting from any arbitrary state.) However, we do not make any assumptions about the activities that the character can do in regards to anticipation. Instead, we add a large variety of anticipation clips (*examples*) to a database, including actions such as taking protective steps, ducking, or lifting a leg off the ground to protect against a threat. In contrast to previous research [Metoyer et al. 2007] which builds a model of anticipation drawn from psychology, we rely on human performance in the form of anticipation examples to produce life-like anticipatory actions and focus our effort on the construction

of a reliable method for automatically selecting which example to employ based on a trained model of how to minimize damage and expend the least amount of energy in doing so.

2 Related Work

Several researchers have introduced techniques that generate responses for motion capture-driven characters reacting to unforeseen influences [Oshita and Makinouchi 2001; Zordan and Hodgins 2002; Yin et al. 2003; Shapiro et al. 2003; Mandel 2004; Zordan et al. 2005; Arikan et al. 2005; Komura et al. 2005]. In general, previous methods for responding to an interaction take into account the physical components related to the impact, either in the form of a simulated collision, as in [Faloutsos et al. 2001; Zordan and Hodgins 2002; Mandel 2004; Zordan et al. 2005], or by modifying dynamic parameters of the character motion, such as joint velocities [Oshita and Makinouchi 2001; Arikan et al. 2005] or momentum [Komura et al. 2004; Komura et al. 2005]. The result of creating these kinds of changes is character motion that gives the impression of responding physically following the impact.

To our knowledge, our previous research effort [Metoyer et al. 2007] is the only one reported on anticipation for character response. In that paper, we use insights drawn from psychological literature to infer the proper behavior mechanisms to employ for anticipatory action and develop heuristic rules that are consistent with psychological findings. Our previous results included anticipating impacts to the head and upper body. In contrast, here, we employ examples of anticipation taken from motion capture which are more natural-looking. In addition, we introduce a novel mechanism for learning the conditions leading up to an interaction based on a measure of expected damage and use supervised learning to choose the best anticipation action for a given situation.

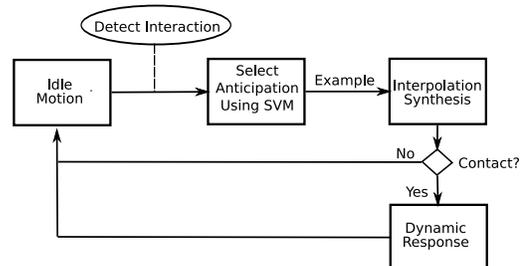


Figure 1: System flow diagram

3 Overview

Our system combines a selection routine, which decides the anticipation action to employ from a library of examples, with an interpolation synthesis step which blends from the current motion to the anticipation motion, taking into account balance. The selection routine uses a Support Vector Machine (SVM) to choose the best anticipation action based on information provided about the upcoming interaction. In an offline process, we train the SVM to classify using a specialized performance index which weighs the expected damage against the amount of energy expended. At runtime, the SVM classifies the input conditions and passes the anticipation motion example clip to the interpolation synthesis process

*Contact author: vbz@cs.ucr.edu

for balance adjustment and blending. Following the system flow diagram in Figure 1, we assume the character starts in idle standing balance (upper left.) When an interaction (or threat) is recognized, the system selects the anticipatory clip to employ using the SVM. Next, in the interpolation synthesis step, a transition to the example motion is generated, making adjustments to balance by controlling the center of mass across the transition. After the anticipation is complete, we compute a dynamic response [Zordan et al. 2005] if there is contact and finally return to the idling motion with a final blend.

4 Assessing damage

One important question we raise in this research is what is the functional effect of anticipatory action. Unlike Metoyer et al. [2007], we attempt to answer this question as a means for assessing *how* to anticipate a given scenario automatically. We do this by quantifying the effect an anticipation has on a given interaction according to a predicted damage assessment. To assess the success of a particular anticipation from our database for a given scenario, we simulate a version of the interaction, including the anticipation to be tested, and measure the amount of “damage” the character sustains. By simulating several such *what-if* scenarios, we can compare a number of anticipation actions for a given test interaction. In doing so, we can make a conclusive statement about the best anticipation action to take from our anticipation database according to our damage assessment. In practice, it is impractical to run many simulations before selecting the anticipation, especially in an interactive setting, such as a virtual environment or electronic game. Instead, we use this simulation process to produce a training set over which to learn the bounds of specific characteristics (features) that will make a particular anticipation successful over another.

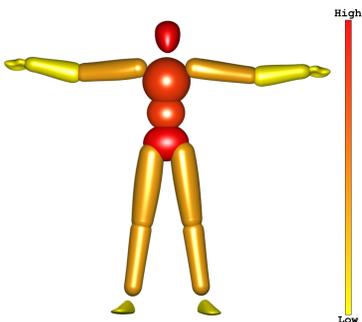


Figure 2: Vulnerability diagram by body part

The proposed damage metric is a key element in the success of our approach. In particular, we assess damage based on the amount of impulse applied to different body parts, where the sensitivity to damage is assigned as a weight according to the vulnerability diagram shown in Figure 2. Based on this weighting, a small to moderate impulse to the lower arm will cause less “damage” than similar or larger impulse applied to the chest. In many cases, contact is made with several parts of the body and each contribute to the amount of damage assigned to a given scenario. While damage alone is enough to rank anticipation actions, it does not differentiate between a dramatic leap and a subtle lean which both may prevent contact from a potential blow. Thus, in addition to damage, we add a second term to our assessment which accounts for the amount of work or energy needed to achieve a given anticipation. With this term, the metric can differentiate the lean from the leap and would give preference to the lean if the damage were comparable (because the work/energy required would be much less.)

To relate the raw damage and energy terms, we introduce a single weighting value, h , which we refer to as the “brawn” factor because

it dictates how much pain a character is willing to sustain, based on how much effort (work) the character is willing to expend to avoid that pain. And we can summarize our damage assessment metric by combining the terms as such:

$$damage = \sum_{i \in \mathbf{B}} ((1-h)v_i I_i + hW_i) \quad (1)$$

where $\mathbf{B} = \{1, \dots, m\}$ for m bodies, v_i is the vulnerability rating (Figure 2.) Each body’s impulse, $I = \int F_{external} dt$, is computed for all simulated contact forces felt by that body in the specific simulation of the given interaction. We compute the mechanical work for each body, $W = \int F_{internal} dS$, from the motion trace S taken from the anticipation example clip and $F_{internal}$ based on body mass estimates times acceleration computed numerically from the motion trace. While this term is not an accurate measure of the actual energy exerted by the human, it is a very reasonable indicator of the relative amount of energy present in each example motion. Note, the work terms only need to be computed once for each action in the database. The brawn h is used to normalize the maximum expected value of the work term to that of the impulse term. Also, this factor provides a single intuitive number for the animator to tune. We set the value to 0.1 for all of our results in this paper.

5 Supervised learning

We employ a Support Vector Machine (SVM) [Boser et al. 2002] to classify the conditions leading up to an interaction and select the specific anticipation action from our database. The SVM works by training in an offline process and then performs selection quickly during runtime. We use SVM to select among possible anticipation actions based on specific conditions (or features) sampled when an impending interaction is (supposedly) recognized by the character. Our SVM is built from training data that binds the features of an interaction with the best anticipation example to employ according to our damage metric.

Intuitively, SVM works by finding partitions in the space of input data. Formally, SVM is used to fit functions which maximize the error margin between samples found in a training set. Let $\mathbf{T} = \{(x_1, \alpha_1), \dots, (x_l, \alpha_l)\}$ be a set of training observations where $x_i \in R_n$ are the samples of the feature vector with n dimensions and α_i are the corresponding anticipatory actions (labels) chosen to be best by exhaustive search. The SVM is created by solving the following optimization problem [Boser et al. 2002]:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^\top w + C \sum_{i=1}^l \xi_i \\ \text{subject to} \quad & \alpha_i (w^\top \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

Training examples of feature vectors, x_i , are mapped to a multi-dimensional space with separating partitions found with maximal margins by varying weights, w , and probability estimates, b . Cost $C > 0$ is a weighting penalty for errors terms ξ_i . $\phi(x_i)^\top \phi(x_j)$ is defined to be kernel function $K(x_i, x_j)$ and, for our purposes, we use polynomial kernel $K(x_i, x_j) = (\gamma x_i^\top x_j + t)^d$, $\gamma > 0$ with γ, t, d as user-defined kernel parameters. In our implementation, we employed the library LibSVM (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>) which includes useful defaults for the user-defined terms.

We choose SVM over other approaches, such as K-nearest neighbor (KNN), because of the speed of prediction given the complexity of our training vector. While KNN can be sped up using spatial partitioning, the larger the dimension the more difficult it is to partition effectively (in order to be efficient in finding close neighbor examples.) SVM, in contrast, finds the partition automatically and, at run-time, is very fast to compute - which is important for real-time

applications such as games and virtual environments. In addition, Faloutsos et al. [2001] employ SVMs for character animation in the training of preconditions for various behavior controllers and while his problem is different, he reports higher accuracy for SVM over KNN for his application.

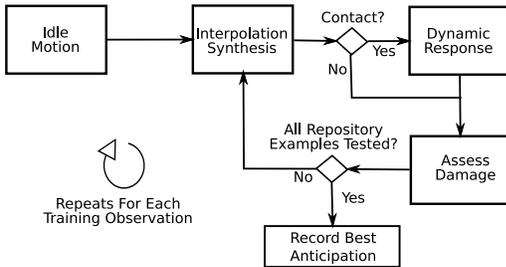


Figure 3: Training flow diagram

Training. For each observation added to the training set we follow the flow diagram in Figure 3. That is, to compute a single observation the system simulates the test interaction in turn for each motion example in the anticipation repository, assessing the damage metric and keeping track of the best action to take based on the minimum damage from all the simulations. We run this process from random starting points, i.e. by varying the threat and character state for each test interaction. While it is too slow to run such an exhaustive search (including many simulated interactions) online, this training is done offline and only once per database.

Our feature vector is a hand-selected set of characteristics taken from the test interaction at the time the character ‘recognizes’ the threat. In particular, we include the feature attributes which describe the state of the threat at the time of recognition, namely its relative position, heading, and speed. In addition, in the results in this paper, we provide two additional pieces of information that we use as input for initializing the ball threat: the expected time to impact and the target body part. While this information could be computed by predicting the ball’s path of travel and finding the intersection between that trajectory and the character’s movement, we preempt this prediction and provide the data directly to ensure the information is accurate. Finally, we encapsulate the state of the character in the feature vector simply, with a lookahead of the global position and facing direction for the predicted time of contact.

6 Balance Adjustment

To generate animation given the anticipation action, we propose a simple but effective interpolation synthesis technique that accounts for balance. The problem here is to concatenate the motion the character is currently following with the anticipation motion in the example. To be successful, the transition should not introduce any unwanted artifacts. The most straightforward solution is to align the anticipation motion globally to the character’s current position and facing direction and then to blend the root position and orientation as well as the joint angles. However, this would likely introduce unnatural foot sliding since there is no guarantee that the feet are in the same configuration before and after the transition.

To overcome this issue, we apply a balance adjustment step. Our specific routine is somewhat related to previous approaches for balance filtering [Tak et al. 2000; Metoyer et al. 2007] but is unique because we approximate a purposeful weight shift. That is, the system shifts the weight to one foot, based on the foot in the anticipation example that is carrying more of the weight. Then, we use the newly selected “support foot” as the fixed root position for blending and use the inverse kinematics (IK) routine described by Metoyer

et al. on the other leg to keep the foot on the ground. The goal is to make the character lean in the direction of one foot and slide the other foot into place - the effect is a quick adjustment of weight that is largely imperceptible and, we believe, quite natural for situations where double support is followed by a sudden anticipatory movement.

To accomplish balance adjustment, our algorithm moves the center of mass smoothly toward the support foot, running an optimization step (BFGS) to place the pelvis, and using IK to reposition the legs while keeping the feet fixed. With the center of mass over the support foot, the system performs the blend to the selected example by aligning the support foot in the example motion with the current motion. During blending, the center of mass is returned smoothly to the unmodified position for the anticipation example. While we do not move the non-support foot explicitly in the balance adjustment process, the weight is shifted to the foot which is used as the (fixed) root for the interpolation blend, and the non-support foot is therefore allowed to move as the blend takes place. We found it important for visual quality to use IK on the non-support leg and keep the foot on the ground, both avoiding lifting the foot and from potentially passing through the ground.

7 Implementation and results

To realize our approach, we had to make several engineering and design decisions including how to construct our database and how to implement the various components. For our anticipation library, we include a set of examples which we capture methodically as blocks and anticipations protecting from several directions, targeting various areas of the body (legs, pelvis, trunk, head) and with two varying degrees, mild and exaggerated. In the capture, the subject was asked to imagine a threat approaching from each of eight directions. To ensure that the directions were consistent, a second person stood outside of the capture region and acted out a throwing motion to cue the subject. After each direction, the subject reset to a *home* position and the ‘thrower’ moved to the next location in preparation for the next direction. In total, we include sixty examples of anticipation which are segmented and mirrored (left-to-right) totaling 120 in the final repository.

Our observation set for training the SVM included over 8200 examples to produce pleasing results - computed in about 60 cpu hours on modern processors. The simulation of interactions for training calculated impulses using a tracking controller [Zordan and Hodgins 2002] and an ODE physical simulation (see www.ode.org) with ODE’s collision handler. Even with this many examples, infrequently the SVM would choose undesirable results. To minimize pathological anticipations such as a lucky dodge of the head which results from a block using the legs, we apply a mask to the training data which limits the anticipations that are tested to those targeting a specific body part or its immediate neighbor.

To create the animations shown in the video, our system interpolates to the anticipation example by ‘slerp’-ing quaternions, with a simple ease-in/ease-out (EIEO) time-based weighting across the transition. With the blended motion to the anticipation in place, the interaction itself is finally computed. If the interaction results in a contact (e.g. the SVM did not choose an anticipation that dodges a ball completely,) we incorporate a version of Zordan et al.’s [2005] technique for responding to unpredicted impacts. This subsystem utilizes the ODE simulation to react physically to collision forces and then generates a smooth transition by interpolating between the anticipation and simulated motion. After a short duration, the simulation is blended back to the idle behavior or, at the animator’s discretion, to a reaction example as described in the original implementation.

Results. We show in the accompanying video a variety of examples (See Figure 4) where the character is able to successfully dodge an

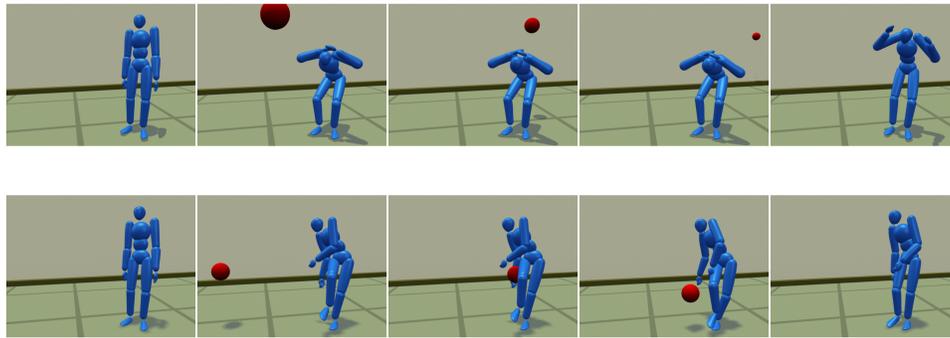


Figure 4: Animation filmstrips. Two examples of anticipation from our system (view left to right).

incoming ball as well as believably anticipate and physically react to threats which the character is unable to avoid completely. In addition we show a set of animations where we make other modifications to test the limits of the system, namely: we modify the starting state to begin from a fighting motion taken from a different source (actor); and we add an example where the dynamic response returns to a new reaction example. The dynamic response is too slow for real-time currently, but without it, the system runs interactively using an AMD Athlon64 CPU with 2 Gigabytes of memory (with hardware rendering.)

8 Conclusions

In this paper, we present an approach for generating anticipation using human motion capture examples. We employ supervised learning to select the example based on the given scenario and train our learner on observations where damage and energy are factors in determining the most suitable anticipation for the conditions of the scenario. To limit our scope, we focus our attention on a specific testbed where a standing character responds to a threat approaching from a variety of trajectories.

As described, there are a number of limitations with our current approach. Foremost, we assume that the character is starting in standing balance and while we do allow the character's state to vary, the range of starting states for most of our animations is fairly narrow. This issue is due to the fact that the anticipation we generate is computed using interpolation synthesis and to generalize to a larger set of starting states we would need to improve this component of the system in order to uphold the quality of the motion generated. Regardless, our contributions for damage assessment and fast selection using SVM will likely be beneficial for more general systems. In addition, in some animations, the character appears omnipotent or "super-human." Adding delays, noise, and/or failed attempts at anticipation would fix this problem and fit nicely within our existing framework. Finally, we point out that the vulnerability rating will need to change based on the threat, for example comparing a ball to a knife, the expected sensitivity of different body parts is drastically different.

Even with these limitations, this work represents a large step forward in the state of the art for automatically generating anticipation action for characters and we look forward to further advances in this exciting topic in the near future.

References

- ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2005. Pushing people around. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM Press, New York, NY, USA, 59–66.
- BOSER, B., GUYON, I., AND VAPNIK, V. 2002. A training algorithm for optimal margin classifiers. In *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM Press.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *SIGGRAPH '01*, ACM Press, New York, NY, USA, 251–260.
- KOMURA, T., LEUNG, H., AND KUFFNER, J. 2004. Animating reactive motions for biped locomotion. In *VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology*, ACM Press, New York, NY, USA, 32–40.
- KOMURA, T., HO, E. S., AND LAU, R. W. 2005. Animating reactive motion using momentum-based inverse kinematics. *Compute Animation and Virtual Worlds 1*, 16, 213–223.
- MANDEL, M., 2004. Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis. *Master's Thesis, Carnegie Mellon University*.
- METOYER, R., ZORDAN, V. B., HERMENS, B., WU, C. C., AND SORIANO, M. 2007. Psychologically inspired anticipation and dynamic response for impacts to the head and upper body. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*. to appear.
- OSHITA, M., AND MAKINOCHI, A. 2001. A dynamic motion control technique for human-like articulated figures. *Computer Graphics Forum 20*, 3.
- SHAPIRO, A., PIGHIN, F., AND FALOUTSOS, P. 2003. Hybrid control for interactive character animation. In *Pacific Graphics 2003*, 455–461.
- TAK, S., YOUNG SONG, O., AND KO, H.-S. 2000. Motion balance filtering. *Computer Graphics Forum 19*, 3 (August), 437–446.
- YIN, K., CLINE, M. B., AND PAI, D. K. 2003. Motion perturbation based on simple neuromotor control models. In *Pacific Graphics*, 445–449.
- ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 89–96.
- ZORDAN, V. B., MAJKOWSKA, A., CHIU, B., AND FAST, M. 2005. Dynamic response for motion capture animation. *SIGGRAPH '05: ACM Trans. Graph. 24*, 3, 697–701.