Predicting General Motion of Humanoids using Supervised Learning

Adriano Macchietto and David Brown

October 25, 2013

Introduction

Predicting humanoid motion has many interesting applications. This project was primarily motivated by the desire to create an artificial virtual game AI that can accurately target and evade a fully-articulated humanoid character. Due to the emergence of human motion interface devices, such as the Microsoft Kinect 3D camera system, players now have the ability to control their avatars through full body movement. To maintain a sense of immersion for the gamer, having artificial agents that can predict, learn, and adjust to the players motion input in a realistic manner is important. This project attempts to address this problem.

We build a supervised learner which can accurately predict the generalized mass motion of a humanoid. Specifically, we show that through use of support vector machines we can predict the movement of the center of mass (CM) for different rigid body chains of the character. Since we concern ourselves with large horizon windows, we do not attempt to predict the full-body dynamics of the character. Instead, we focus on the more general mass motion of the character that changes more slowly and predictably. For many applications, such as targeting the character with a projectile, this is sufficient.

To qualify our results we use a truncated Taylor series predictor. We demonstrate that not only is our SVM predictor more accurate than the Taylor series predictor, but also more scalable for large prediction horizons.

Feature Extraction

To obtain our dataset we first recorded a series of motion capture clips designed to represent a wide range of motion styles. We use three main categories: stationary, locomotion, and evasive motions. For the stationary motions the actor kept his feet planted on the ground at all times. We recorded the actor bending over to touch his toes, twisting his upper body, and attempting to move his body as much as possible while standing in place. For the locomotion motions we had the actor walk in a circle, jog in a line, skip, do a hopscotch motion, and perform jumping jacks while moving forwards. Finally, for the evasive motions had the actor perform evasive motions on his own then we recorded two motions where another person tossed projectiles at

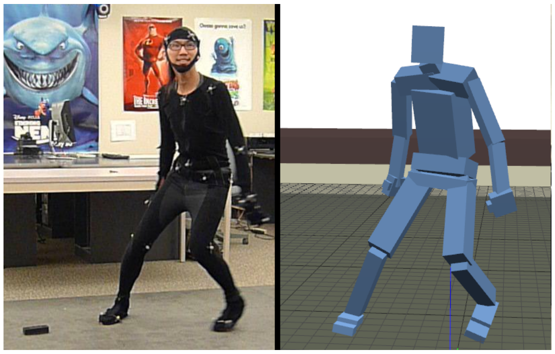

Figure 1: (left) A live actor evading a series of projectiles. (right) The actors physical mass model.

the actor as the actor attempted to evade them. We use one of these dodging motions as our primary testing motion since it is an example of how humans attempt to dodge projectiles in practice. All our motions are between three and ten seconds long, and we chose a roughly equal number of motions for each category.

We chose these specific motions so our predictor will have a good variety of examples with which to learn various types of human motions. The stationary examples force the agent to keep his center of mass relatively consistent, showing how the extremities can move while the center of mass position remains fairly constant. They also train the predictor to be robust for stationary agents.

The various locomotion examples are designed help the predictor characterize walking, running and hopping motions. These motions are very regular and train the system to predict motions with consistent velocity and momentum. However, since our motion capture space is limited in size we could only record short locomotion examples with relatively slow speeds. This issue manifests itself when we are performing predictions on the jogging example. The jogging example is our fastest moving example motion and our predictor tends to predict behind the agent's actual position since it has no faster examples to learn from.

The evasive examples are intended to train the predictor on how an agent moves when trying to actively avoid being hit. This shows the robustness of our system in the face of very complex motions. These three categories cover a wide range of human motion, however we did limit our data set to motions where the feet are the primary support, so our predictor will likely only make good predictions where this is the case.

A physical model of the actor is also included with the motion capture data. This gives us dynamic information allowing us to use the physical properties of the motion as features instead of just the kinematic trajectories. Our motion capture data is filtered to smooth out any large acceleration jumps and remove any small errors in the motion capture data. This smoothing looks at both previous and future frames to smooth the motion which does add information from a couple of frames in the future to the present frame. However, since our frame rate is 120HZ and we are predicting over very large horizons this has a negligible effect on our results.

From this motion capture data we extracted several features to characterize the agent's full body motion. Since our goal was to predict over large horizons we generalized the motion capture data using key features that would help with these predictions.

One obvious, but vital, feature is the horizon length so the predictor knows how far ahead to predict. When extracting the training data from our motions, we randomly choose a horizon length over a specified interval. We generated the best results when we centered this interval about the horizon length we wanted to predict. Another important feature is the agent's full-body center of mass. This feature is implied in our data because all positional features are given relative to the agent's full-body center of mass. We do this because the future positions of the agent is very directly based on the agent's current full-body CM.

Linear and angular momentum, and their time derivatives, are also very fundamental features in describing an agent's motion. The linear momentum tells us where the agent is moving and how that movement is changing. Angular momentum is important because humans induce angular momentums for very dynamic motions such as dodging, and we also regulate angular momentum to help maintain balance. We also include the CM and linear and angular momentum of various rigid body chains of the agent. We calculate the center of mass C, linear momentum, P, its derivative $\dot{P}$, angular momentum L, and its derivative, $\dot{L}$, of a set of bodies B as shown:

$$C(B) = \frac{\sum_{i \in B} m_i \mathbf{x}_i}{\sum_{i \in B} m_i}, \tag{1}$$

$$P(B) = \sum_{i \in B} m_i \dot{\mathbf{x}}_i, \quad \dot{P}(B) = \sum_{i \in B} m_i \ddot{\mathbf{x}}_i, \tag{2}$$

$$L(B) = \sum_{i \in B} I_i \omega_i, \quad \dot{L}(B) = \sum_{i \in B} I_i \dot{\omega}_i \tag{3}$$

where m_i, I_i, $\mathbf{x}_i$, and ω_i are the mass, the inertia tensor, the CM, and the angular velocity respectively.

These features allow us to better predict where different rigid body chains will be in the future and they can also help with predicting the full body motion of the agent. For example the position of the agent's feet will limit the type and direction of motions the agent can make.

Feature Set	**Feature**
A	horizon length; full-body momentum
B	horizon length; full-body momentum; head CM
C	horizon length; full-body momentum; leg CM
D	horizon length; full-body momentum, head, arm, leg, thorax CM
E	horizon length; full-body momentum; leg momentum
F	horizon length; full-body momentum; head momentum
G	horizon length; full-body momentum; arm, leg, head, thorax momentum
H	horizon length; full-body momentum; arm, leg, head, thorax CM; arm, leg, head, thorax momentum

Table 1: Descriptions of the various feature sets we tested with the system

	Motion					
	Squat		Hopscotch		Dodge	
Data Set	Mean (m)	Std. (m)	Mean (m)	Std. (m)	Mean (m)	Std. (m)
Stationary	0.0683	0.0468	0.4900	0.1634	0.2878	0.1049
Locomotion	0.1645	0.1282	0.1929	0.0814	0.1972	0.1100
Evasion	0.8700	0.0601	0.3848	0.1533	0.1719	0.1129
All	0.0200	0.0235	0.1021	0.0669	0.1522	0.1234

Table 2: This table shows the accuracy of the SVM predictor as a function of the test motion and training dataset used. In all cases, the prediction accuracy is best when the tested motion is trained on a dataset of similar motions. A significantly lower overall mean error is achieved by combining the categorized training datasets into a single dataset.

Supervised Predictor

We used support vector machines as our machine learning approach. Originally, we had attempted to use neural networks, but due to the slowness of convergence and lukewarm preliminary results we switched to SVM. The Libsvm library [1] is used as our SVM implementation.

Libsvm provides two SVM regression algorithms, ϵ-SVR and ν-SVR. After much experimentation, we could not find any significant performance difference between the two algorithms. ν-SVR was arbitrarily picked. The model parameters used for all results were chosen by random sub-sampling validation with a 20%/80% train/test split on all training data. The validator selected a Gaussian kernel with a width of .2, a margin penalty constant of .2, and a ν parameter of .5.

Results

We compared our SVM to a simple 1st and 2nd-order truncated Taylor series predictor. Let x, t and Δt denote the predicted CM, time, and the horizon window length. The first and second order truncated Taylor series predictors are given by

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t \tag{4}$$

and

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2 \ , \tag{5}$$

respectively. The future value of the predicted feature is computed from the first and second order derivatives of the motion data. Since our feature coordinates are a function solely a function of the joint coordinates, the feature derivatives are a function of the joint coordinates, velocities, and accelerations. Because motion

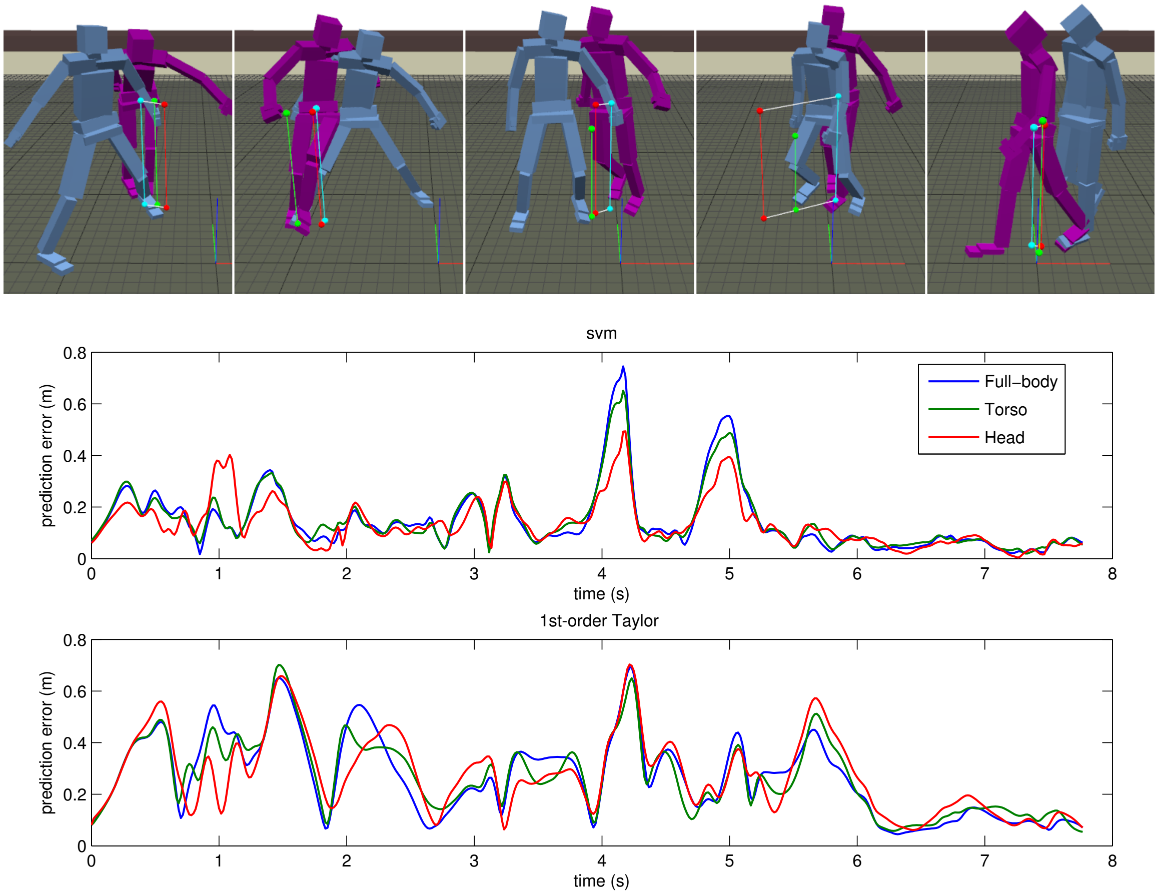

Figure 2: (top) Samples from the *dodge* test motion for time 0.7, 1.9, 3.1, 4.1, $7s$. The blue and purple characters denote the current and future posture ($.5s$ horizon). The blue, red and green spheres are the future full-body CM, the full-body CM prediction by the SVM, and the full-body CM prediction by the 1st-order truncated Taylor series predictor, respectively. (center, bottom) Mean distance error vs. time for the SVM predictor and 1st-order truncated Taylor series predictor across different mass groups.

Feature Set	**Predicted Feature (CM)**							
	Full Body		Head		Right Arm		Right Leg	
	Mean (m)	Std. (m)	Mean (m)	Std. (m)	Mean (m)	Std. (m)	Mean (m)	Std. (m)
A	0.1622	0.1211	0.1799	0.0987	0.3254	0.2013	0.2435	0.1556
B	0.1613	0.1214	0.1586	0.1004	0.3285	0.1959	0.2280	0.1701
C	0.1591	0.1197	0.1556	0.0892	0.2984	0.1881	0.1996	0.1593
D	0.2423	0.1513	0.1831	0.1154	0.3682	0.2242	0.2882	0.1961
E	0.1476	0.1247	0.1443	0.0742	0.3023	0.2128	0.2251	0.1799
F	0.1588	0.1207	0.1701	0.1110	0.3230	0.2045	0.2420	0.1599
G	0.1608	0.1227	0.1841	0.1083	0.2966	0.2192	0.2217	0.1856
H	0.1457	0.1282	0.1369	0.0995	0.2842	0.2040	0.2061	0.1708

Table 3: Output feature error as a function of the selected input features (see Table 1)

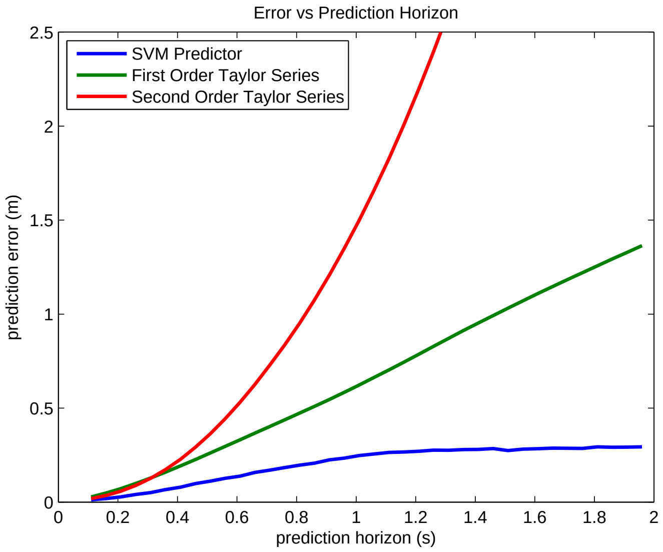

Figure 3: Performance comparison of the various predictors as a function of the horizon window. The SVM predictor scales better with than the first and second-order Taylor predictors. The leveling-off of the error is likely due to the limitations of the motion-capture space being reflected in the trained motion (for many motions the actor was required to walk in circles to stay within the permissible capture region).

capture only provides a sequence joint coordinates, we approximate the velocities and accelerations of the joints through central finite differencing. A low-pass filter is then applied to remove any high-frequency noise introduced during reconstruction of the marker data. Due to the 120HZ sampling rate of the motion-capture system, we limit ourselves to first and second-order Taylor predictors only.

Figure 2 presents the difference between the the SVM predictor and the truncated Taylor series predictor. The distance error between the predicted feature value and the true feature value is calculated each frame of the *dodge* test motion. The SVM predictor was trained on all motions excluding *dodge*. Here we can see that the SVM predictor performs significanly better with a mean error and standard deviation of .149m and .128m, compared the 1st-order Taylor predictor with .269m and .155m, respectively.

In general the SVM predictor performs well, except during the 4 and 5 second mark. At both times, the actor uses a feigning strategy which involved taking a large step forward only to use the new support foot to push backwards and quickly reverse all forward momentum to fool the adversary throwing the projectiles. Out of all motions tested, this was the most difficult motion to acurately predict; Both predictors strongly assume continued forward momentum. Given the large prediction horizon of .5, features such as the projectile direction or the projectile emitters motion, may be needed to better predict how the actor will attempt to evade.

Typically, the system was quite accurate for non-evasive motions. Table 2 summarizes the test performance across different training sets.

Figure 3 demonstrates the the SVM scales better with the horizon length. On might notice that the error seems to curiously level off past the 1s mark. This is due to the limited size of the motion capture region which requires the character to circle around for locomotive tasks, thereby limiting the maximum possible displacement of the CM in all training examples.

Conclusion

We have shown that it is possible to accurately predict aggregate mass features for full-body human characters across large horizons using simple features. In general, the system is able to make far better predictions

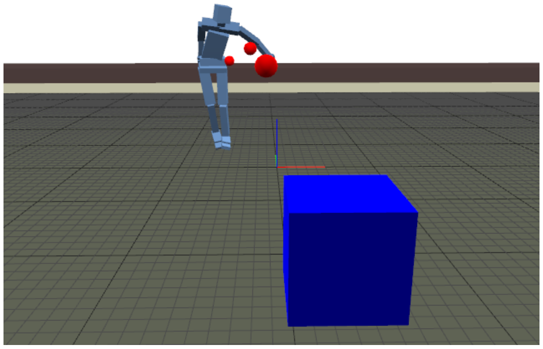

Figure 4: Projectiles shot from an emitter at the predicted future CM of the character.

than the more naive unsupervised method. In addition, we have shown that the system scales better with the prediction window than the Taylor series predictor. Although the system works very well for regular periodic motions, and admirably for motions with irregular momentum changes, it was easily fooled in cases where the actor was actively shifting momentum irregularly in order to deceive the system.

Future proposed modifications to the system may include features which capture more of the cognitive decision-making processes (head orientation, threat direction, and threat dynamics), rather than just the dynamic limitations of the character.

Acknowledgements

We like to thank Nam Nguyen for providing us with his motion.

References

[1] Chang, C.-C., and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.