

A Responsiveness Metric for Controllable Characters

Technical Report CS05-50-01

Madhusudhanan Srinivasan * Ronald A. Metoyer †
School of Electrical Engineering and Computer Science
Oregon State University

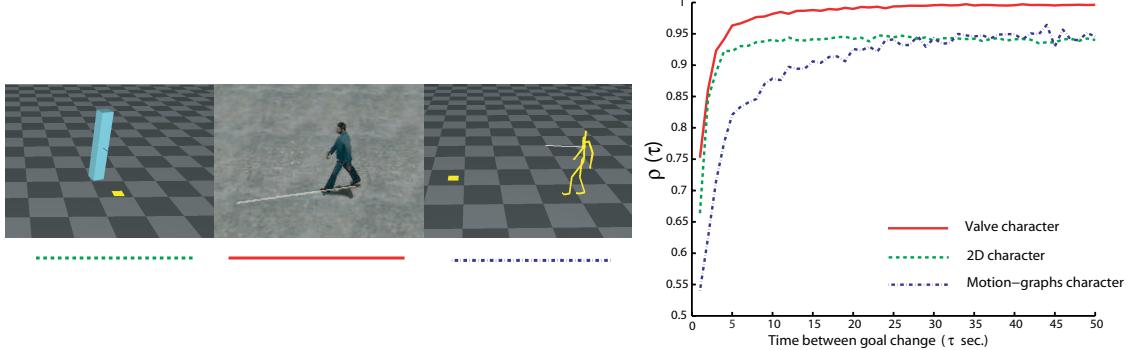


Figure 1: Quantifying and comparing responsive behavior for three different techniques. Images from left to right: 1. 2D locomotion using a coarse motion-graphs based approach. 2. Character locomotion using an interactive game SDK. 3. Character locomotion using a motion-graphs based approach. 4. A comparison of Reactivity Characteristic Curves for the three techniques. A steeper curve indicates a more responsive character. Each technique involves a character tracking a spatial target on the floor.

Abstract

This paper presents an empirical approach for measuring and characterizing the responsiveness of a character to changes in goal. Our approach is based on keeping track of the character’s progress towards a frequently changing goal. A “distance-to-goal” function is defined to measure the progress. We then calculate an asymptotic proportion of the progress made towards the goal. Plotting this proportion for different frequencies of goal changes produces a responsiveness characteristic curve for the goal directed character. We use the responsiveness characteristics to compare three interactive locomotion approaches. We illustrate how the metric can be used to tune and optimize the responsiveness of a character.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Animation, Interactive Avatars, Motion Graphs.

1 Introduction

One of the main goals of character animation is to produce believable and controllable characters for 3D interactive environments.

*e-mail: srinivma@eecs.oregonstate.edu

†e-mail:metoyer@eecs.oregonstate.edu

Many recent techniques have produced characters that can be intuitively controlled by specifying a goal such as a target pose or a desired spatial location. While populating an interactive environment with directible characters, a content creator must choose the most appropriate technique. Among other factors, the choice may depend on both the quality of the motion and the ability to direct or control the character. Recently, Ren et al. developed a metric that quantifies the quality of human motion [Ren et al. 2005]. Unfortunately, there are no known metrics for quantifying the degree of control over the character. Such a metric could help a content creator select the motion generation technique that is most suitable for his or her interactive application. For example, consider the creation of a pedestrian evacuation simulator. For such an environment, the user may require that the 3D motion be able to closely track the data from a 2D simulation while maintaining natural transitions to create compelling 3D motion. The user may evaluate the naturalness of different approaches using the metric introduced in [Ren et al. 2005]. However, there are no reliable metrics available to evaluate controllability of these approaches.

The responsiveness of a character may depend on several parameters. For instance, in a motion-graphs based approach, responsiveness may depend on the motion dataset, transition thresholds and transition cost formulation. As Reitsma and Pollard pointed out, there are open questions regarding how much motion data should be included in a motion graph and what parameter values should be chosen to make the character sufficiently responsive to user inputs [Reitsma and Pollard 2004]. To answer these questions, we need a metric that measures the responsiveness of a character.

In this paper, we introduce a metric for evaluating the responsiveness of a goal-driven character to a frequently changing goal. Our focus is on locomotion techniques that direct a character toward a spatial target (goal) on the floor. In an environment where the goal changes periodically (at a fixed rate), we define responsiveness to measure how quickly the character responds to changes in the goal. We define a function that measures the “distance” of the character

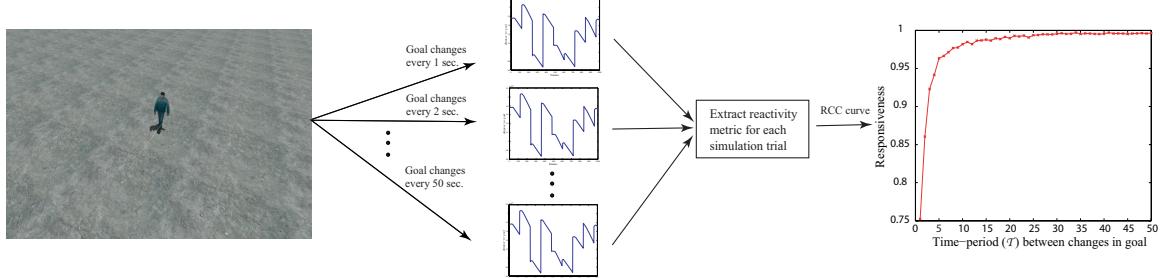


Figure 2: An overview of our approach to quantify responsiveness of a directible character. Our experiment consists of multiple trials. Each trial changes the goal to a random spatial location on the floor, at a different rate.

from its goal. We then compute a ratio of the total decrease in the value of the distance function to the total change in the value of the function. If the character is responsive, we show that this ratio converges to an asymptotic value. We then define the value to be the *responsiveness* of the character. Since the controllability of a character is characterized by its response to changes in the goal, we define responsiveness at a particular rate of change of the goal. A *responsiveness characteristic curve* (RCC) is constructed by plotting the responsiveness of the character at different rates of change of the goal.

We validate our metric by computing RCC plots for three different locomotion techniques and comparing them (Figure 1). We also demonstrate the use of the metric for tuning parameters. We finally demonstrate the use of RCC plots in determining the effect of including different motion datasets in a motion-graph based approach.

The rest of the paper is organized as follows. We begin with a review of related work. We provide a brief outline of our approach in section 3. In section 4, we describe our experimental setup to collect data required to estimate responsive behavior. In sections 5 and 6, we describe our approach to compute the responsiveness characteristic curve. Finally, we conclude with results and discussion.

2 Background

A considerable amount of research has been done in the area of generating controllable character animation. Early work focused on using procedural techniques and physics based controllers to simulate and control animated characters [Kuffner 1998; Perlin 1995; Hodgins et al. 1995]. Inverse Kinematics has also been used as a technique to control the pose of a character [Wiley and Hahn 1997; Grochow et al. 2004]. Introduction of motion graphs to generate automatic transitions between motion clips was a significant contribution for synthesizing controllable character motion from captured motion data [Kovar et al. 2002; Arikan and Forsyth 2002; Lee et al. 2002; Gleicher et al. 2003]. Lee et al. precompute character response to user inputs and control character motion at run time [Lee and Lee 2004]. Other techniques for character control include using annotations and examples [Arikan et al. 2003; Hsu et al. 2004; hoon Kim et al. 2003]. Parameterized Interpolation and blending techniques have also been used to generate controllable character motion [Rose et al. 1998; Park et al. 2002; Kovar and Gleicher 2003; Kovar and Gleicher 2004].

Recent efforts have focused on developing tools that quantify and evaluate different aspects of motion generation techniques. Wang and Bodenheimer analyzed different cost metrics for defining transitions between motion segments [Wang and Bodenheimer 2003].

Their method was used to find an optimal set of joint weights for a motion-graphs cost function. More recently, Wang and Bodenheimer developed an empirical approach for determining the proper duration of motion transitions [Wang and Bodenheimer 2004].

Reitsma and Pollard introduced a metric for evaluating path quality and coverage of motions generated from a motion graph [Reitsma and Pollard 2004]. They introduced a new algorithm to embed the entire motion graph into the environment in order to capture the area that can be covered by the graph. More recently, Ren et al. presented a metric that quantifies the naturalness of human motion [Ren et al. 2005].

To our knowledge, the present work is a first attempt to measure the quality of user control over the character. We focus on measuring how quickly a character responds to changing spatial user requests and we call this the responsive locomotion behavior of a character. While our work shares the above goals of developing a metric to evaluate motion generation techniques, it is different in the sense that it applies to measuring the quality of user control, not the quality of the motion itself. Our metric is not limited to a specific type of motion generation technique and can be applied to any goal directed method.

3 Overview

Figure 2 shows an overview of our approach. To measure responsiveness, we consider an environment in which the target goal is changing. Our experiment consists of simulating multiple trials with different rates of goal change. As each trial progresses in time, we track and record the progress of the character to the goal using a distance-to-goal function. We use this statistic to estimate the responsiveness of the character for a particular rate of change of goal. A Responsiveness Characteristic Curve (RCC) is then plotted that quantifies the character’s responsive behavior for various frequencies of goal change.

4 Responsiveness Trials

For each trial, our environment consists of a 3D character on a 2D floor plane. We assume that the underlying motion generation technique provides for the character to track target locations (goal) on the floor plane. We also assume that we can run the trial with rendering turned off. This enables us to collect a large amount of data in a relatively short period of time.

A trial consists of generating random goal locations on the floor plane. The frequency at which the goal location changes is repre-

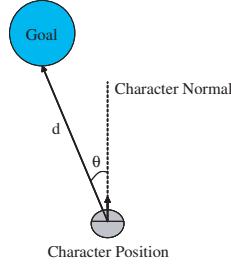


Figure 3: Distance between the character and its goal is measured in terms of the euclidean distance $d(t)$ and the deviation angle $\theta(t)$.

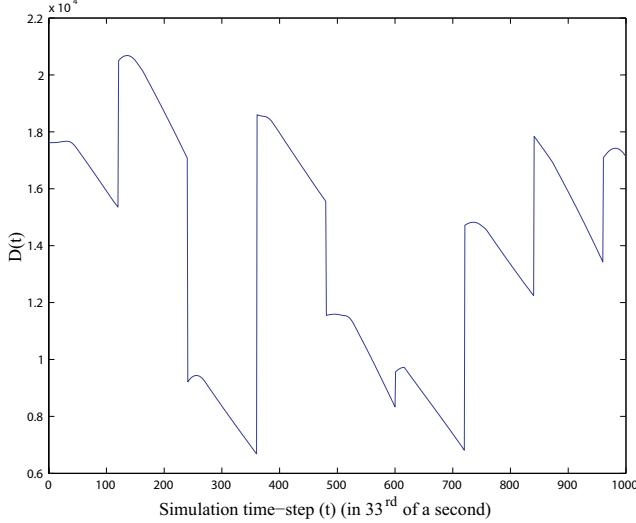


Figure 4: A plot of $D(t)$ for a trial that lasts $T = 1000$ frames. The goal changes every $\tau = 5$ seconds or every 165 frames (1 sec = 33 frames).

sented by a parameter, τ . A new random goal is generated every τ seconds. Trial time is measured in frame units (every 33rd of a second). A trial is simulated for T frames.

For our particular experiments, responding to a change in goal position involves turning, possibly changing speed, and ultimately walking towards the new goal. This motivates our definition of distance between the character and the goal. The distance $D(t)$ between the character and its target at any time t (measured in terms of frames) is defined as

$$D(t) = \omega_d \cdot d(t) + \omega_\theta \cdot \theta(t) \quad (1)$$

where $d(t)$ is the euclidean distance (in meters) between the target location and the character's projected location on the floor plane, and $\theta(t)$ is the deviation angle in degrees. The deviation angle is measured as the angle between the character's facing direction and the vector to the target from the character (Figure 3). For our simulations, we chose $\omega_d = 17.0$ and $\omega_\theta = 1.0$. It is to be noted that $D(t)$ is just a metric that measures the “closeness” of a character to the goal for a locomotion behavior. A different “closeness-to-goal” metric would be used for behaviors other than locomotion.

Figure 4 shows a plot of $D(t)$ for a sample trial with $T = 1000$ frames and $\tau = 5$ seconds. Points of discontinuities in this plot correspond to the instants in time when the goal was moved. Changes

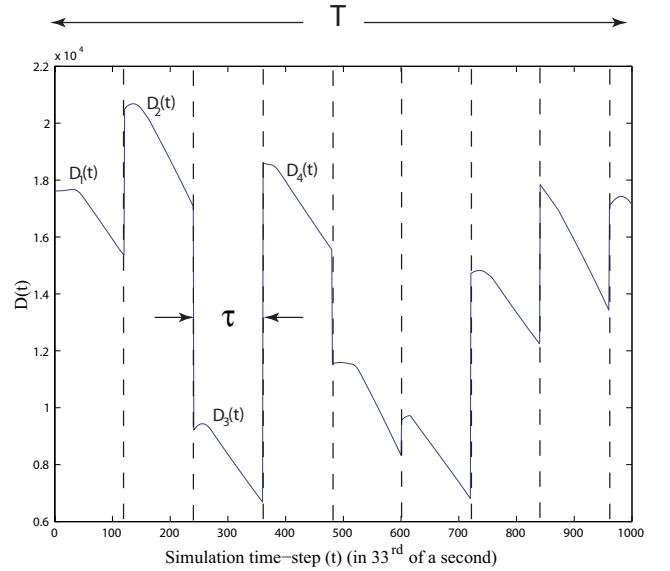


Figure 5: $D(t)$ is represented as a set of $T/(\tau \cdot 33)$ continuous functions, each defined over an interval of length τ seconds.

in goal position cause discrepancies in estimating $D(t)$ and should not be considered. For this reason, $D(t)$ is broken into a set of continuous functions where each function is defined over an interval of τ seconds (or $\tau \times 33$ frames) (Figure 5). In other words

$$\dot{D}(t) = \{\dot{D}_1(t), \dot{D}_2(t), \dot{D}_3(t), \dots, \dot{D}_K(t)\} \quad (2)$$

where $K = T/(\tau \times 33)$. Our experiment consists of simulating trials for different values of τ ranging from 1 to 50 seconds. As we shall see in the following section, the accuracy of the responsiveness estimate depends on the duration, T , of each trial.

5 Estimating Responsiveness

Consider the plot for $D(t)$ in Figure 5. A positive slope implies an increasing value of distance metric and corresponds to a character moving away from the target. A negative slope implies a decreasing distance metric, and the character is progressing towards the target. Given a plot of $D(t)$ ($0 < t \leq T$) for a particular τ , we compute the change in $D(t)$ at every frame. Since $D(t)$ is a discrete valued function, we compute the rate $\dot{D}(t)$ using finite differences.

$$\dot{D}(t) = \{\dot{D}_1(t), \dot{D}_2(t), \dot{D}_3(t), \dots, \dot{D}_K(t)\} \quad (3)$$

Figure 6 shows a plot of $\dot{D}(t)$ for $D(t)$ in Figure 5.

A positive value of $\dot{D}(t)$ corresponds to the character moving away from the target and a negative $\dot{D}(t)$ corresponds to the character moving closer to the target. We can thus partition the function $\dot{D}(t)$ into two separate functions $\dot{D}^+(t)$ and $\dot{D}^-(t)$.

$$\dot{D}^+(t) = \{\dot{D}_i(t) : \dot{D}_i(t) \in \dot{D}(t), \dot{D}_i(t) > 0\} \quad (4)$$

$$\dot{D}^-(t) = \{\dot{D}_j(t) : \dot{D}_j(t) \in \dot{D}(t), \dot{D}_j(t) \leq 0\} \quad (5)$$

Our next step is to estimate the total distance traveled towards the goal. Consider a trial of duration T with goal changing every τ

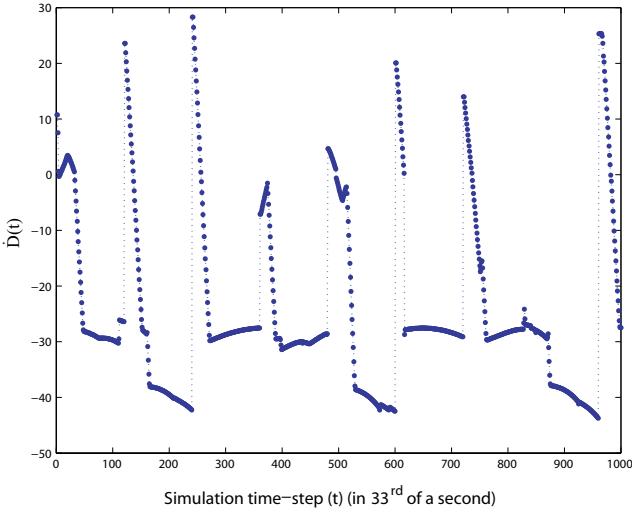


Figure 6: Computing the change in $D(t)$ (Figure 5) at every time-step yields a plot for $\dot{D}(t)$.

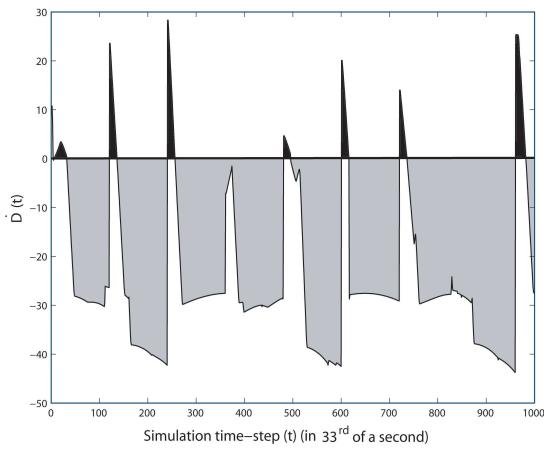


Figure 7: $\Delta^-(T, \tau)$ is the total area of the gray region in the above plot of $\dot{D}(t)$. $\Delta^+(T, \tau)$ is the total area of the dark region above the zero line.

seconds. The total distance traveled towards a goal, $\Delta^-(T, \tau)$, is then computed as the total area under $\dot{D}^-(t)$ (Figure 7).

$$\Delta^-(T, \tau) = \sum_{\forall \dot{D}_j(t) \in \dot{D}^-(t)} \left\{ \int \dot{D}_j(t) dt \right\} \quad (6)$$

This distance is in terms of the metric defined in Equation 1. Similarly, the total distance traveled away from the target is computed as the area under $\dot{D}^+(t)$.

$$\Delta^+(T, \tau) = \sum_{\forall \dot{D}_i(t) \in \dot{D}^+(t)} \left\{ \int \dot{D}_i(t) dt \right\} \quad (7)$$

Since $\dot{D}(t)$ is a discrete valued function, the continuous integrals are approximated by summations.

We now define a responsiveness function $R(T, \tau)$ as the ratio of distance traveled towards the target ($\Delta^-(T, \tau)$) to the total distance

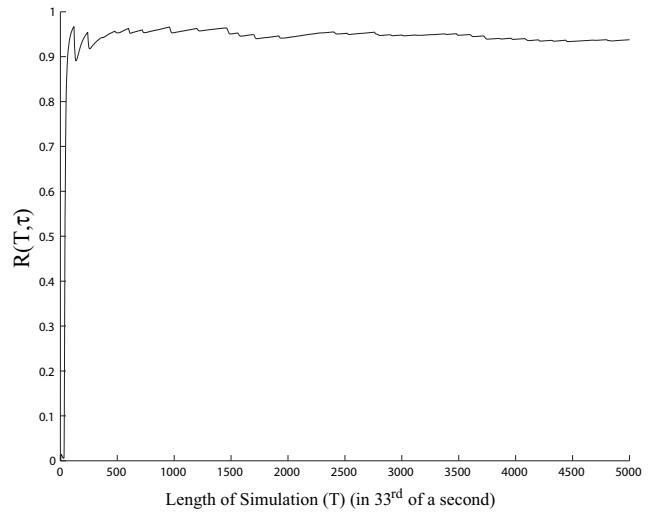


Figure 8: A plot of $R(T, \tau)$ versus length of a simulation trial. $R(T, \tau)$ stabilizes with time.

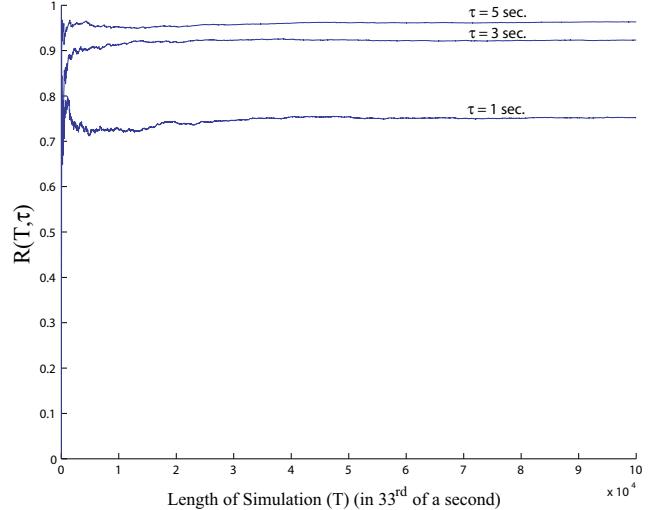


Figure 9: A plot of $R(T, \tau)$ for different values of τ .

traveled by the character during the entire simulation.

$$R(T, \tau) = \frac{\Delta^-(T, \tau)}{|\Delta^+(T, \tau)| + |\Delta^-(T, \tau)|} \quad (8)$$

Figure 8 shows the plot of $R(T, \tau)$, for $\tau = 5s$ and $T = 5000$ frames. Figure 9 shows the plot of $R(T, \tau)$ for different values of τ .

It can be observed that the value of $R(T, \tau)$ stabilizes as we increase the duration T of a simulation trial, for a given τ . We define *responsiveness* $\rho(\tau)$ of a character, for a target that changes every τ seconds, to be

$$\rho(\tau) = \lim_{T \rightarrow \infty} R(T, \tau) \quad (9)$$

For practical purposes, we approximate this limit by running our simulations for about $T = 100,000$ frames, for different values of τ .

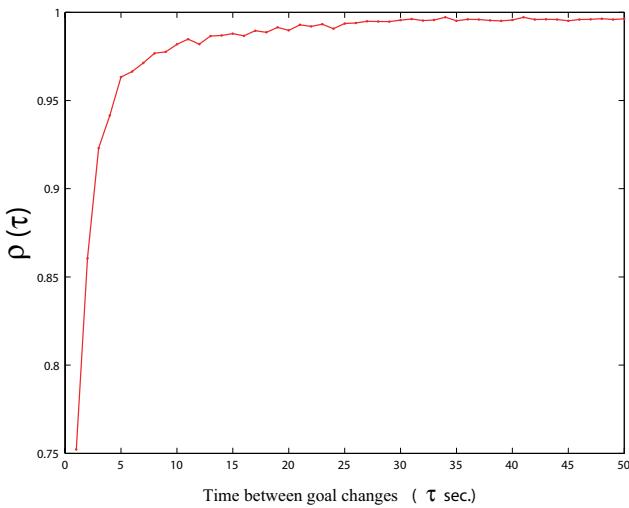


Figure 10: Responsiveness Characteristic Curve (RCC). Any point on the curve corresponds to the responsiveness, $\rho(\tau)$, of the character for a goal changing every τ seconds.

6 Responsiveness Characteristic Curve

Computing and plotting $\rho(\tau)$ for different values of τ (1 to 50 seconds) results in a *Responsiveness Characteristic Curve* (RCC) for the character (Figure 10). Any point on the curve represents the measure of responsiveness of the character for a particular τ . Since $\rho(\tau)$ is measured as a proportion of the total distance, its value always lies in the interval $[0,1]$. In general, a character is less responsive for more frequently changing goals (small τ) and more responsive for less frequently changing goals (large τ). This explains the nature of the RCC plot. The curve rises steeply in the beginning, starting with a low responsiveness value for frequently changing goals. The curve approaches 1.0 as the changes in goal become less frequent. A larger value of $\rho(\tau)$ for a smaller τ indicates a larger proportion of distance traveled towards a more frequently changing goal, and therefore a more responsive character. Hence, a steeper RCC plot corresponds to a more responsive behavior.

7 Results

In the following sections, we present three experiments. In the first experiment, we compute RCC plots for three different character locomotion techniques. The second experiment illustrates parameter tuning for a motion-graph based approach using our responsiveness metric. In the third experiment we compare the responsiveness of a character animated with different motion datasets in the motion-graph.

7.1 Validation

To validate our metric, we construct and compare RCC curves across three character locomotion techniques. The first technique is based on a motion graph similar to that presented by Lee et al.

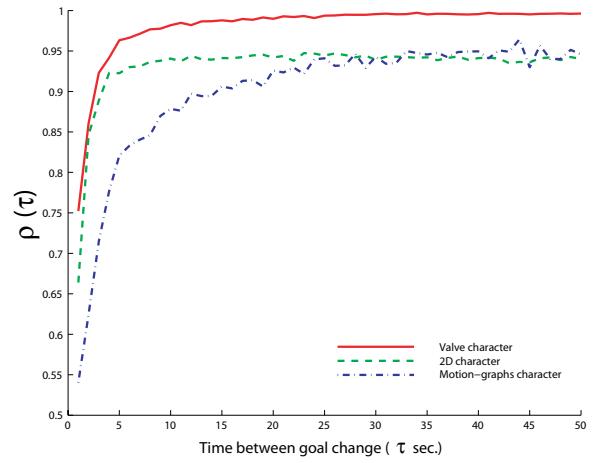


Figure 11: RCC plots for three different locomotion generation techniques. A steeper RCC plot corresponds to a more responsive technique. The Valve SDK character is most responsive.

[Lee and Lee 2004]. The second technique is based on a coarse-grained motion-graph. We group similar poses together before constructing a motion-graph. This produces a more connected graph, although the 3D motion may not be as smooth or natural. We use this clustered graph to produce only 2D locomotion of a low level-of-detail character. The third technique involves using the Valve Software Development Kit (Valve SDK) for character locomotion [Valve 2004]. The SDK is a popular tool used for producing interactive games such as Half-life 2.

We have a qualitative idea of how responsive each of these techniques are. We expect the Valve character to be more responsive than the other two techniques, since the SDK is used in interactive games. However, we expect the motion to be of lower quality because game engines trade off transition quality for fast response to user commands. A clustered motion-graph is dense and well connected. Thus, we expect the responsiveness of a clustered motion-graph technique to be higher than that of the original motion-graph based technique. A plot of RCC curves for these three techniques illustrates and validates our expectations (Figure 11).

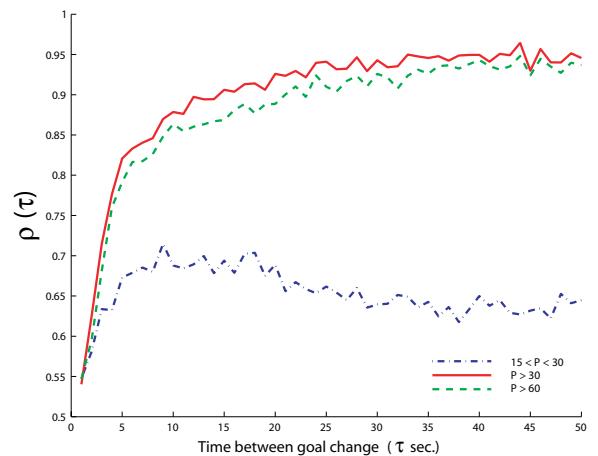


Figure 12: RCC plots for different values of P for a motion-graph. Again, the steepest plot corresponds to the best value of the parameter.

7.2 Parameter Tuning

We have also experimented with using RCC plots to tune the parameters of our locomotion algorithm. We demonstrate by picking different pruning thresholds P for the original motion-graph based technique. Intuitively, a given value of P corresponds to selecting “hub” poses with P outgoing transitions in the motion-graph. We compute the RCC curve for three different values of the parameter (Figure 12). We then choose a threshold that yields the steepest RCC plot to maximize responsiveness.

7.3 Motion Dataset Comparison

In our final experiment, we try to answer the question: What motion sequences should we include in a data driven approach (i.e. motion graphs) to allow the character to respond reasonably to goals in an interactive environment? Reitsma et al. experimented with different motion sets to answer a similar question [Reitsma and Pollard 2004].

For our experiment, we used a library of motion capture data consisting of 114 sequences for a total of 33,404 frames of data. The data was captured with the Vicon 612 3D optical motion capture system [Vicon 2004]. We captured an actor performing various locomotion behaviors such as walking straight at various speeds, turning at various radii, coming to a stop, starting from a stop, standing in place, walking in an imaginary crowded environment, tracking an imaginary target, and turning in place.

We created different versions of a motion-graph using different motion datasets. Our goal was to identify which sequences contributed the most to responsive behavior. Figure 13 shows RCC plots for different motion sets. Table 1 shows the responsiveness measure for these datasets for $\tau = 5$ s. Our experiment reveals that removing crowded walks and tracking does not seem to affect responsiveness. This suggests that we can build a smaller motion-graph without any change in the responsive behavior of the character.

Motion Dataset	Responsiveness ($\tau = 5$ seconds)
Full Set	0.8207
Without Left, Right Turns	0.6836
Without Crowded Walks	0.8289
Without Tracking	0.8418
Without Stop and Turn	0.7276

Table 1: Responsiveness of a motion-graph based character using different datasets.

8 Discussion

We have presented an approach to quantify and characterize responsive behavior of an interactive character. We used interactive locomotion techniques to illustrate the steps in our approach. We also introduced the notion of responsiveness characteristic curves (RCC) that characterize responsive behavior for different frequencies of goal changes. We have presented a qualitative validation of our approach by computing and comparing RCC plots for three different controllable locomotion techniques. We have also presented experiments to illustrate the utility of RCC plots for tuning parameters and to determine how much motion data should be included in a motion graph. Each experiment required approximately 10 minutes for data collection and RCC computation.

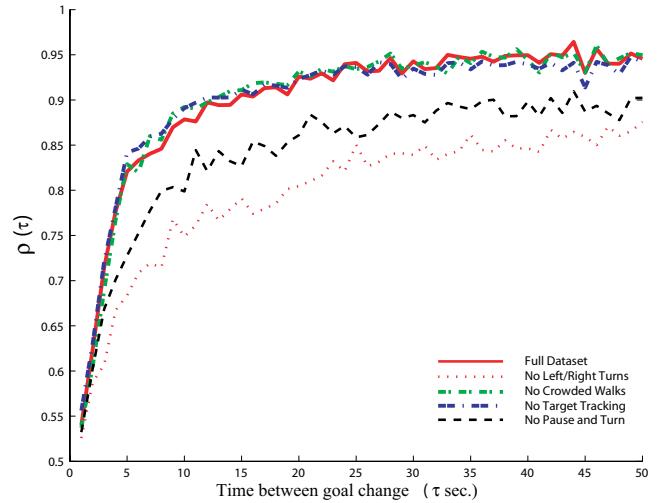


Figure 13: RCC plots for characters using different motion sets. Turning motion contributes the most to character’s responsive behavior. Crowded walking and Target tracking does not improve responsive behavior.

A naive approach for measuring responsiveness would be to measure the time taken for the character to reach a goal. However, we define responsiveness to be the response of a character to a changing goal. The naive metric would not completely characterize responsiveness according to this definition. For instance, a fast walking character may reach a goal in a roundabout way in a short time while a slow walking character may reach the same goal more directly but in a longer time.

It is important to note that our metric is for comparing motion generation techniques under the same conditions. For example, if we compare the responsiveness of a walking character to a running character, our metric would assign a lower responsiveness measure to the running behavior, since a running character requires more time to respond to goal changes in a believable manner.

Our responsiveness metric is based on a generic measure of “distance-to-goal”. This allows us to define responsiveness for tasks other than locomotion control. For example, consider a character controlled by annotated target poses such as “dive”, “catch”, “run” etc. Our distance-to-goal measure for such a character could be the number of poses between the current pose and the target pose. We could use the same steps we illustrated to compute the RCC plots for such a character. In this case, responsiveness would be measuring how quickly the character responds to changes in the user’s task specification (run, dive, etc.).

In our implementations, we randomly select goals over a finite floor plane. However, we can also selectively choose goals that identify specific motion data required to increase the responsive behavior of a character. We can specifically measure a character’s responsive behavior by specifying goals in a particular direction with respect to the character. A small responsiveness value may suggest the type of motion data required. For example, goals always to the right may help identify if one has enough right-turn data.

The generality of our metric allows us to build and compare the responsiveness characteristics of different techniques for the same character behavior. For instance, consider soccer animations. A soccer player’s path is affected by many factors such as defenders, game strategy and physical strength of the player. A soccer player may be animated using several techniques such as a procedural ap-

proach or a motion-graphs based approach. Responsiveness trials under identical experimental conditions can be carried out for both of the techniques to compute and compare the RCC plots. Although our approach would measure the responsiveness of the techniques for generating controllable soccer motion, it does not measure the quality of the soccer motion. To assess the naturalness quality of the motion, our metric could be used in conjunction with the naturalness metric introduced by Ren et al. [Ren et al. 2005]. For instance, a player that simply teleports to the soccer-ball location is perfectly responsive in our formulation, however, such an animation may not score well in terms of the naturalness metric.

One of the main drawbacks of our approach is that we need a large amount of data (distance-to-goal at every frame) to get an accurate estimate of responsiveness. In our experiments, we were able to collect data in a short period of time by turning off rendering and simulating character response at high frame rates (≈ 1000 fps). However, such an option may not always be available to the user. Deciding the amount of data necessary to get a reasonable estimate of responsiveness is another issue. We collected data over approximately 100,000 frames. However, we noted that when the character is sufficiently responsive, the responsiveness metric converges faster. This suggests that we could have computed a reasonable estimate with a smaller set of data.

References

- ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics* 21, 483–490.
- ARIKAN, O., FORSYTH, D., AND O'BRIEN, J. 2003. Motion synthesis from annotations. *ACM Transaction on Graphics* 22, 3, 402–408.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPoulos, D. 2001. Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, 251–260.
- GLEICHER, M., SHIN, H., KOVAR, L., AND JEPSEN, A. 2003. Snap-together motion: assembling run-time animations. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM Press, 181–188.
- GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIC, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graph.* 23, 3, 522–531.
- HARRISON, J., RENSINK, R. A., AND VAN DE PANNE, M. 2004. Obscuring length changes during animated motion. *ACM Trans. Graph.* 23, 3, 569–573.
- HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proceedings of ACM SIGGRAPH 95*, Addison Wesley, R. Cook, Ed., Computer Graphics Proceedings, Annual Conference Series, 71–78.
- HODGINS, J., O'BRIEN, J., AND TUMBLIN, J. 1998. Perception of human motion with different geometric models. *IEEE Transactions on Visualization and Computer Graphics* 4, 4, 307–316.
- HOON KIM, T., PARK, S. I., AND SHIN, S. Y. 2003. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Trans. Graph.* 22, 3, 392–401.
- HSU, E., GENTRY, S., AND POPOVIC, J. 2004. Example-based control of human motion. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press.
- KOVAR, L., AND GLEICHER, M. 2003. Flexible automatic motion blending with registration curves. In *In Proceedings of Eurographics/SIGGRAPH Symposium on Computer Animation (2003)*.
- KOVAR, L., AND GLEICHER, M. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.* 23, 3, 559–568.
- KOVAR, L., GLEICHER, M., AND PIGHIN, F. H. 2002. Motion graphs. *ACM Trans. Graph.* 21, 3, 473–482.
- KUFFNER, J. 1998. Goal-directed navigation for animated characters using real-time path planning and control. In *CAPTECH '98: Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, CAPTECH '98, 171–186.
- LEE, J., AND LEE, K. H. 2004. Precomputing avatar behavior from human motion data. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press.
- LEE, J., CHAI, J., REITSMA, P., HODGINS, J., AND POLLARD, N. 2002. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics* 21, 491–500.
- METOYER, R., AND HODGINS, J. 2000. Animating athletic motion planning by example. In *Proceedings of Graphics Interface*, 61–68.
- METOYER, R., 2002. Building behaviors with examples. Ph.D. Dissertation, Georgia Institute of Technology.
- OESKER, M., HECHT, H., AND JUNG, B. 2000. Psychological evidence for unconscious processing of detail in real-time animation of multiple characters. *Journal of Visualization and Computer Animation* 11, 2, 105–112.
- PARK, S., SHIN, H., AND SHIN, S. 2002. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 105–111.
- PERLIN, K. 1995. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics* 1, 1, 5–15.
- REITSMA, P. S. A., AND POLLARD, N. S. 2003. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM Trans. Graph.* 22, 3, 537–542.
- REITSMA, P. S. A., AND POLLARD, N. S. 2004. Evaluating motion graphs for character navigation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 89–98.
- REN, L., PATRICK, A., EFROS, A. A., HODGINS, J. K., AND REHG, J. M. 2005. A data-driven approach to quantifying natural human motion. *ACM Transactions on Graphics (SIGGRAPH 2005)* 24, 3 (Aug.).
- ROSE, C., COHEN, M., AND BODENHEIMER, B. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5, 32–40.
- VALVE, 2004. Valve software development kit.
<http://www.valvesoftware.com>.

VICON, 2004. Vicon 612 optical motion capture system.
<http://www.vicon.com>.

WANG, J., AND BODENHEIMER, B. 2003. An evaluation of a cost metric for selecting transitions between motion segments. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 232–238.

WANG, J., AND BODENHEIMER, B. 2004. Computing the duration of motion transitions: an empirical approach. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 335–344.

WILEY, D. J., AND HAHN, J. K. 1997. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, 156–160.