An Information-Based Dynamic Extrapolation Model for Networked Virtual Environments

XIN ZHANG, TOMÁS E. WARD, and SÉAMUS MCLOONE, National University of Ireland Maynooth

Various Information Management techniques have been developed to help maintain a consistent shared virtual world in a Networked Virtual Environment. However, such techniques have to be carefully adapted to the application state dynamics and the underlying network. This work presents a novel framework that minimizes inconsistency by optimizing bandwidth usage to deliver useful information. This framework measures the state evolution using an information model and dynamically switches extrapolation models and the packet rate to make the most information-efficient usage of the available bandwidth. The results shown demonstrate that this approach can help optimize consistency under constrained and time-varying network conditions.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*; H.1.1 [**Models and Principles**]: Systems and Information Theory—*information theory*; I.6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies*; I.6.8 [**Simulation And Modeling**]: Types of Simulation—*distributed*

General Terms: Measurements, Performance, Theory

Additional Key Words and Phrases: Consistency, collaborative virtual environments, distributed interactive applications, distributed interactive simulation, networked multi-player computer games, networked virtual environments, information management techniques

ACM Reference Format:

Zhang, X., Ward, T. E., and McLoone, S. 2012. An information-based dynamic estrpolation model for networked virtual environments. ACM Trans. Multimedia Comput. Commun. Appl. 8, 3, Article 27 (July 2012), 19 pages.
DOI = 10.1145/2240136.2240140 http://doi.acm.org/10.1145/2240136.2240140

1. INTRODUCTION

With the rapid development of computation and networking technologies, Networked Virtual Environments (NVEs) have evolved to a group of software systems that allow multiple participants to collaborate and interact with each other simultaneously in a shared virtual environment, even if they are from geographically distant separate locations. Significant deployments of NVEs include distributed military simulations (e.g., SIMNET [Calvin et al. 1993; Miller and Thorpe 1995], DIS [IEEE 1998], HLA [IEEE 2000]), academic virtual networked communities (e.g., DIVE [Frécon and Stenius 1998],

This work is supported by the Irish Research Council for Science, Engineering, and Technology (IRCSET) and funded by the National Development Plan.

Authors' addresses: X. Zhang, T. E. Ward, and S. McLoone, Department of Electronic Engineering, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland; email: {xzhang, tomas.ward, seamus.mcloone}@eeng.nuim.ie

DOI 10.1145/2240136.2240140 http://doi.acm.org/10.1145/2240136.2240140

NPSNET [Capps et al. 2000]), and large-scale online entertainment systems (e.g., Ultima-Online [Origin Systems 1997] and Quake [Kushner 2002]).

Generally, each human-user in an NVE controls a virtual entity (or avatar) through a host machine to interact with others in the simulated virtual environment [Delaney et al. 2006]. The entities are represented by a number of state variables (position, velocity, etc.). To maintain a consistent shared view of the environment, which is vital for meaningful interactions among the participants, the local controlling host periodically notifies the state changes of the corresponding entity by sending synchronization messages (or update packets) across the underlying communication network to the remote host machines. However, the data transmission is subject to finite network bandwidth and non-zero network latency, which make absolute consistency impossible [Capps and Stotts 1997; Roehle 1997]. If the data is being transmitted at a rate that exceeds the available bandwidth of the link between the local and remote host machines, the packets will be queued or even dropped, causing significant increases in network latency and packet loss. In this case, the remote host has to maintain its version of the virtual world with out-of-date state information and suffer from severe inconsistency. This "Consistency-Throughput Trade-off" indicates that it is impossible to achieve a highly consistent and dynamic world and low bandwidth usage at the same time [Singhal and Zyda 1999]. In this article, we focus on the entity state distribution of NVEs where the avatar motion plays a central role and the periodic *state updates* constitute the major part of the network traffic.

To deal with the tradeoff, one important group of consistency maintenance mechanisms, collectively known as Information Management (IM) techniques have been proposed [Singhal and Zyda 1999; Delaney et al. 2006; Marshall et al. 2008]. These techniques introduce a form of controlled inconsistency by reducing the message transmission rate in order to minimize latency caused by an overloaded network. On the remote hosts, the intermediate entity motion between two updated states is extrapolated using some user behavioral model (notable examples include polynomial dead reckoning (DR) [IEEE 1998; Pantel and Wolf 2002b] and advanced statistical modeling techniques [Pantel and Wolf 2002b; Hanawa and Yonekura 2006; McCoy et al. 2007]). In spite of the inconsistency introduced by the reduced number of updates, the reduction in network load should in theory result in reduced latency, which should in turn improve the overall consistency.

However, IM techniques could have a negative impact on consistency maintenance unless carefully adapted to the underlying network conditions and the participating user behavior. Existing works only focus on either one of these two factors. For example, adaptive extrapolation models have been proposed [Lee et al. 2000; Delaney et al. 2003; McCoy et al. 2005], in which an active extrapolation equation that best suits the specific user behavior is dynamically chosen from a pool of candidates solely based on application layer criteria such as local prediction error or object's goal. Typically, different extrapolations require various types of data (such as entity position and velocity) to be transmitted. But the influence of changing the extrapolation model on the network traffic and, in turn, the overall consistency is not considered. In contrast, in order to optimize consistency, Marshall et al. [2008] tune the data transmission rate of an IM technique to match the available bandwidth. In this way, the bandwidth usage is maximized to control the inconsistency introduced by insufficient updates, and in the meantime the inconsistency caused by increased latency and packet loss on an overloaded network is avoided. However, the maximized bandwidth usage alone cannot guarantee an optimized consistency without considering the suitability of the prediction model for extrapolating the specific user behavior. None of the existing works combines both application and network layer factors in optimizing consistency.

In our previous work, a new approach to examine consistency maintenance mechanisms from the perspective of information theory has been introduced. This method measures the amount of information carried by the state derivatives in an update message that can be used to extrapolate remote entity motion [Zhang et al. 2008, 2009]. Concepts such as Entropy and Mutual Information are employed to examine the user behavior from the perspective of predictability and to measure the efficacy of the network traffic generated in delivering such predictability to the remote model. The advantage of such information-based measurement is that it does not require the assumption of a user behavior model or pattern, and provides a general metric of the user motion predictability. The inconsistency caused by the reduced updates and latency is modeled as information loss.

This work presents the design of a novel Information-Based Dynamic Extrapolation Model for IM techniques. Using the information model as performance evaluation, the dynamic extrapolation model first examines the capacity of the transmitted data to encapsulate information about the user behavior, based on which the extrapolation method and the update packet rate are dynamically adjusted by the local host such that IM techniques can achieve the maximum usage of the available network bandwidth, and in the meantime deliver as much information as possible about the true entity motion to the remote host. Such operations are transparent to the remote host. Instead of proposing a more accurate prediction model, the purpose of this work is to demonstrate a novel information perspective to regulate consistency maintenance in NVEs, which enables the proposed model to combine both application and network layer factors when adjusting IM parameters and to provide a more thorough optimization of IM techniques. Results collected from experimental studies on recorded user motions from representative NVE scenarios demonstrate that the proposed approach can accurately switch between extrapolation models and modify the packet generation rate to minimize inconsistency, even with changing network conditions.

The remainder of this article is organized as follows. In Section 2, IM techniques are analyzed using an information model, and the motivations for the proposed technique are demonstrated. The operation of the dynamic extrapolation model is detailed in Section 3. In Section 4, the dynamic extrapolation model is demonstrated through experiments on two user motions that are representative of typical entity movement in NVEs, and the results show that the proposed approach can appropriately switch between extrapolation models to minimize inconsistency. Finally, Section 5 offers conclusions and discusses implications.

2. AN INFORMATION MODEL FOR IM TECHNIQUES

The information model for IM techniques [Zhang et al. 2008, 2009] provides a quantified metric of the temporal dependence within the entity state that makes future states predictable from the current motion status. This metric is independent of any presumed motion model. For the convenience of the reader, concepts in information theory and the information model are briefly introduced. All the definitions and methods here are given in discrete terms, as state variables of all entities in a virtual environment, however vivid, are finite and discrete.

2.1 Information Theory Concepts

Consider a random variable X with m possible states $\{x_1, x_2, \ldots, x_m\}$, each with probability $p(x_i)$. The uncertainty of X is defined by the entropy $H(X)$ of the variable [Cover and Thomas 2006], as shown in (1):

$$H(X) = -\sum_{i=1}^{m} p(x_i) \cdot \log p(x_i). \tag{1}$$

Entropy measures the degree of complexity and unpredictability of the variable. In an intuitive illustration, entropy is the “quantity of surprise when a measurement specifies one particular value x_i”. If the binary logarithm is used, $H(X)$ indicates the average number of binary questions needed to determine the value of X, namely the data in bits to describe the variable. Consider an extreme case where X is completely determined to be x_i, that is, $p(x_i) = 1$ and all other probabilities are zero. We have $H(X) = 0$, which means no extra data is required. On the other extreme, if X is equiprobable with a uniform probability $p(x_i) = 1/m$ for all possible states, it is impossible to predict this completely random variable since no state is more probable than the others. We have the maximal entropy $H(X) = \log m$.

For two interdependent variables X and Y with possible states $\{x_1, x_2, \ldots, x_m\}$ and $\{y_1, y_2, \ldots, y_n\}$ respectively, knowing Y may also give some knowledge about X to reduce its uncertainty. The remaining uncertainty of X after knowing Y is defined as conditional entropy:

$$H(X|Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i, y_j) \cdot \log p(x_i \mid y_j), \tag{2}$$

where $p(x_i, y_j)$ is the joint probability and $p(x_i \mid y_j)$ is the conditional probability. The difference in entropy of X before and after knowing Y denotes the predictability about X provided by Y. Thus, the mutual information $I(X;Y)$ is defined as

$$I(X;Y) = \sum_{x_i, y_j} p(x_i, y_j) \cdot \log \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)} = H(X) - H(X \mid Y). \tag{3}$$

Mutual information measures the dependence between X and Y and how much information about X can be learned from Y so that less data are required to determine X. The unconditional entropy $H(X)$ is also known as auto-mutual-information, namely $I(X;X) = H(X)$.

To apply information theory to NVEs, the probability functions must be estimated from discrete data. It should be noted that there exist sophisticated probability estimation algorithms, such as adaptive partitioning methods [Fraser and Swinney 1986] and Kernel Density Estimation algorithms [Moon et al. 1995; Steuer et al. 2002]. However, the accuracy of such algorithms depends largely on careful selection of parameters such as sample space partitioning and kernel functions, which is a non-trivial process that requires significant research efforts. For the purpose of this paper to demonstrate the concept of using information metrics in optimizing IM techniques, a simple approach [Steuer et al. 2002] is employed. In sampled data $x(k)$ and $y(k)$ of size N, we have

$$\hat{p}(x_i) = \frac{r_i}{N}, \qquad \hat{p}(x_i, y_j) = \frac{r_{ij}}{N}, \tag{4}$$

where r_i is the number of cases that $x(k) = x_i$, and r_{ij} is the number of occurrences that $x(k) = x_i$ and $y(k) = y_j$ at the same time. An additional correction term has to be applied to (3) to eliminate the "Finite-size Effect" [Steuer et al. 2002]:

$$I(X;Y) \approx \sum_{x_i, y_j} \hat{p}(x_i, y_j) \cdot \log \frac{\hat{p}(x_i, y_j)}{\hat{p}(x_i) \cdot \hat{p}(y_j)} - \frac{m_{xy} - m_x - m_y + 1}{2N}, \tag{5}$$

where m_x, m_y denote the number of unique state values in $x(k)$ and $y(k)$, and m_{xy} is the number of unique values of $[x(k), y(k)]$ combined as a vector variable. To make a good mutual information estimation, the sample size N must be considerably larger than the number of possible state combinations [Roulston 1999].

2.2 Information Model

The information model [Zhang et al. 2008, 2009] regards each message as an information carrier that delivers information about the true entity motion to the remote host to improve the certainty of the predicted state model. The information comes from the predictability encapsulated in the state derivatives in the message.

For a true entity state motion $\boldsymbol{x}(k) = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots\}$, where $\boldsymbol{x}(k)$ is the entity state (position) at the *kth* simulation step, a message $\boldsymbol{u}(k)$ generated at the simulation tick k generally consists of the current state and possibly additional state derivatives, depending on the extrapolation model under use. For example, in linear extrapolation, an update packet contains the current position and velocity of the object, that is, $\boldsymbol{u}(k) = [\boldsymbol{x}(k), \boldsymbol{v}(k)]$. In IM techniques, a message is responsible for extrapolating states at multiple steps in the future through the extrapolation model. The extrapolation model is typically a deterministic mapping $f(\boldsymbol{u}(k), \tau)$ from the parameters in $\boldsymbol{u}(k)$ to the modeled state $\tilde{\boldsymbol{x}}(k+\tau)$ at τ steps later. To measure the information utilization of the prediction model, mutual information $I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_\tau)$ is defined between the true entity state $\boldsymbol{x}(k)$ and the prediction result $\tilde{\boldsymbol{x}}_\tau$ as

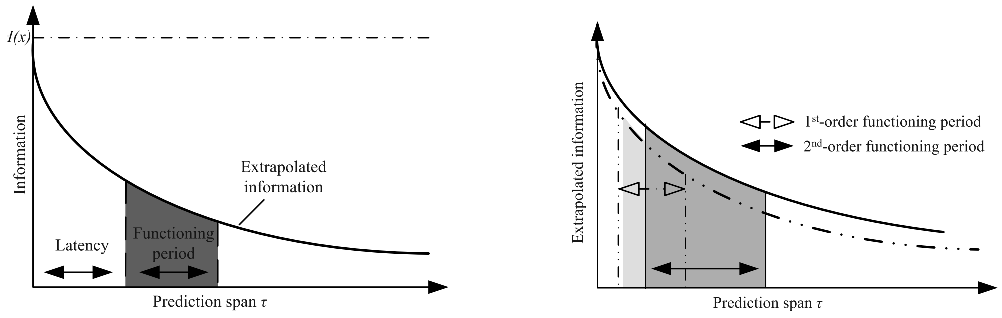
$$\begin{aligned} I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_\tau) &= I(\boldsymbol{x}(k); \tilde{\boldsymbol{x}}_\tau(k)) \\ &= \sum_{\boldsymbol{x}(k), \tilde{\boldsymbol{x}}_\tau(k)} p_{\boldsymbol{x}\tilde{\boldsymbol{x}}_\tau}(\boldsymbol{x}(k), \tilde{\boldsymbol{x}}_\tau(k)) \cdot \log \frac{p_{\boldsymbol{x}\tilde{\boldsymbol{x}}_\tau}(\boldsymbol{x}(k), \tilde{\boldsymbol{x}}_\tau(k))}{p_{\boldsymbol{x}}(\boldsymbol{x}(k)) \cdot p_{\tilde{\boldsymbol{x}}_\tau}(\tilde{\boldsymbol{x}}_\tau(k))}, \\ \tilde{\boldsymbol{x}}_\tau(k) &= \{f(\boldsymbol{u}(1), \tau), f(\boldsymbol{u}(2), \tau) \ldots\} \text{ for } k > \tau. \end{aligned} \tag{6}$$

Notice that the message $\boldsymbol{u}(k)$ acts as an information carrier that encapsulates the predictability of the true entity motion and conveys such predictability to the extrapolation model to build the remote model. The mutual information in (6) measures the average information from the messages utilized by the prediction model in producing the prediction result at a particular prediction span τ. In the case where the prediction span $\tau = 0$, $I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_0)$ is essentially the auto-mutual information, and thus the current state can be determined with absolute certainty. For $\tau > 0$, the entity motion in the future is generally less related to the current motion status. But the deterministic nature of the extrapolation model ignores the potential possibilities of the future states other than the predicted value. Therefore, only part of the uncertainty of the future state to be extrapolated could be eliminated and the extrapolated information declines with increasing prediction spans. To keep the remaining uncertainty in this article a tolerable level, the local host must keep sending update messages to the remote host.

The amount of information utilized is subject to the suitability of the extrapolation model in exploiting predictability in user behavior. This predictability comes from user behavior patterns, real world physics, and environmental constraints. For example, in a car racing game when an object is moving steadily along a straight lane, the information $I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_\tau)$ of the $1st$-order extrapolation is expected to be high because the entity velocity remains the same, and the position and velocity value in the message is sufficient to determine the future entity state using linear extrapolation. In other words, high information value of the extrapolation model means strong predictability and low uncertainty about the estimated state.

Aside from the extrapolation model, network conditions also constrain the utility of the extrapolated information within the functioning period T_f of the message, which is the time interval between two packet arrivals on the remote host. Figure 1(a) illustrates the situation for general NVE scenarios with the information extrapolated from a message for increasing prediction spans. Assuming a fixed latency for the clarity of illustration, the length of the functioning period of a message is the constant packet generation interval of the NVE. Therefore, the average rate of utilized information per simulation time step is obtained by averaging the mutual information $I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_\tau)$ for all the prediction spans within the functioning period, as in (7):

$$MI_u = \frac{\sum_{\tau \in T_f} I(\boldsymbol{x}; \tilde{\boldsymbol{x}}_\tau)}{T_f}. \tag{7}$$



(a) (b)

Fig. 1. Illustration of (a) the information utilization in IM techniques and (b) the information trade-off between packet size and update rate.

As shown in Figure 1(a), the true entity motion always has the perfect information rate $H(\boldsymbol{x})$, but this is reduced to the extrapolated information rate MI_u on the remote host by IM techniques. The down-sampling rate, extrapolation model, and network conditions are the three factors that introduce information loss and cause inconsistency: the imperfect extrapolation model only utilizes a part of the predictability in the user motion; network latency makes the information in the messages outdated by the time they arrive on the remote host; network bandwidth constrains the packet update rate, and a low update rate makes a message suffer from more information decay before the next message arrives. From the information perspective, to maintain consistency using IM techniques, the extrapolated information rate must be optimized under the constraints of these three factors.

In the case where an IM technique uses a fixed extrapolation model, the optimal update rate can be determined such that the overall data transmission rate matches the available network bandwidth, and the functioning period of the messages is reduced to minimize information loss. However, the issue arises when the IM technique can choose from several prediction methods to optimize the delivered information rate. Since continuous entity states that evolve with the passage of time are predictable from contextual dynamic, the impact of a lower update rate could be compensated by including more information in the packet to, possibly, reduce prediction error. On the other hand, a higher update rate could also make up for a reduced amount of information per packet. Consider two popular polynomial equations for extrapolation, namely the linear and $2nd$-order equations. The 2^{nd}-order extrapolation model uses the additional acceleration and is likely to produce higher prediction precision, because physical laws are simulated in most continuous entity movements in NVEs [Singhal 1996; Lee et al. 2000; McCoy 2007]. But the update rate must be reduced accordingly to maintain the optimal data rate under the larger packet size. Figure 1(b) illustrates the situation by comparing the functioning periods of the two extrapolations. Although the $2nd$-order model extrapolates more information than the $1st$-order equation, the extra functioning period due to the lower packet rate and higher latency due to the larger packet size make it suffer more from out-of-date and declined information while the $1st$-order extrapolation has received a new packet with updated and higher information rate. Whether or not the faster update rate of the $1st$-order extrapolation can compensate for the lack of the acceleration information depends on the information decay of the two extrapolations and the locations of their functioning periods. In other words, IM techniques must adapt both packet rate and extrapolation models to the user behavior and network conditions to optimize the overall information rate.

In the next section, an information-based extrapolation scheme that dynamically adjusts both extrapolation model and update packet rate to optimize the remote information rate is discussed.

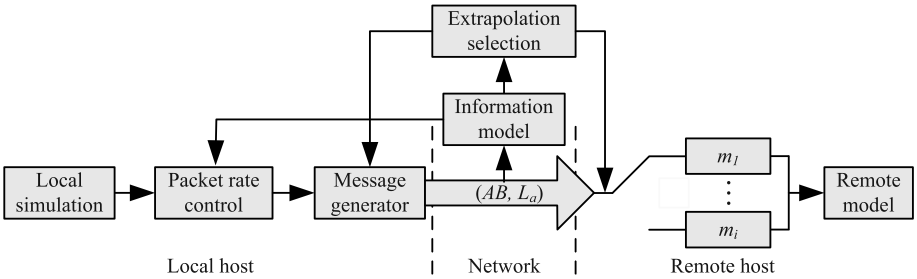


Fig. 2. Illustration of the Information-Based Dynamic Extrapolation Model.

3. OVERVIEW OF THE INFORMATION-BASED DYNAMIC EXTRAPOLATION MODEL

The technique proposed here is designed to optimize overall consistency of a Peer-to-Peer or Client/ Server NVE that uses IM techniques from an information perspective. In such scenarios, the "last mile link" of the users Internet connection is typically the bottleneck that constrains the consistency maintenance the most. This Information-Based Dynamic Extrapolation Model operates by monitoring the bottleneck link bandwidth and end-to-end latency to adjust the packet rate and extrapolation model according to the estimated remote information rate. By dynamically switching the extrapolation model and packet rate, the proposed technique not only maximizes the usage of the available bandwidth, but also results in more efficient data transmission from an information perspective.

The diagram in Figure 2 presents the operation of the Information-Based Dynamic Extrapolation Model. The model is a closed-loop consistency control technique that operates in an *Observation-Feedback-Adjustment* manner. The information model works as the "forward observation" that evaluates performance of the extrapolation models, based on which one best suits the user behavior and the underlying network conditions is selected through the "feedback control" mechanism. The model consists of the following five major components.

(i) A pool of i candidate extrapolation models given by $M = \{m_1, m_2, \ldots, m_i\}$ and state derivatives $U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_i\}$ used for these prediction models, respectively. The set of extrapolation models also defines the size of the update packets $PS = \{ps_1, ps_2, \ldots, ps_i\}$ for each model. A default active extrapolation model m_0 is chosen from M for the application to start with.

(ii) A runtime estimation scheme that monitors network conditions on the application layer's "logical connection" for each user. The network condition estimation can be any algorithm that gives the available bandwidth AB on the bottleneck link and the one-way network latency L_a from the local host to the remote host. This is a relatively independent component and works as an interface of the proposed technique to the underlying network. It should also be noted that since the network conditions are estimated for each logical connection separately, the dynamic extrapolation model operates on a per-user basis. It allows for multiple hosts under heterogeneous Internet to take different strategies to optimize consistency according to their own situations.

(iii) An update rate control based on the available bandwidth. The optimal update rate PR_i for each candidate is determined so that the bandwidth usage is maximized, that is, $PR_i = \frac{AB}{ps_i}$.

(iv) An information model that characterizes information utilization of each extrapolation equation. The information model is the core of the proposed technique, as it examines the capability of each prediction model to extrapolate information about the true entity motion to build the remote model, and how the extrapolated information decays with time. Along with the current network condition estimation and the optimal update rate, the remote information rate $MI_{u,i}$ that each extrapolation model m_i can produce is estimated by (7).

(v) An extrapolation selection based on the estimated information rate chooses whichever extrapolation model that produces the highest information rate under the current network conditions as the current active extrapolation m^*:

$$m^* = \arg\max_{m_i} MI_{u,i}. \tag{8}$$

Ideally, in the face of network condition changes, a switch of the active model will always occur when one of the inactive models is seen to be outperforming the rest. However, there are two concerns that have to be considered for smooth user perception in practice. Firstly, the extrapolation model change is deactivated within a time-out T_O after the previous model switch to avoid excessive switchings in response to transient network condition fluctuations. This can also be done by employing a smoothing operator to the network condition measurement. Second, to avoid a sudden change in consistency level at the moment k_0 of extrapolation model switch, a gradual convergence algorithm described in (9) is defined for a smooth conversion to the new active model:

$$\hat{x}(k_0 + k) = \left(1 - \frac{k}{T_C}\right) \cdot m^-(u^-) + \frac{k}{T_C} \cdot m^+(u^+) \text{ for } k \le T_C, \tag{9}$$

where T_C is the convergence period in simulation steps, $m^-(u^-)$ is the state predicted for the current step $k_0 + k$ by the previous model using the last received message before the switch, and $m^+(u^+)$ is the prediction made by the new active model using the most recently arrived message.

It should be noted that in real NVE systems, critical events (such as car crashing in a cay racing game or explosions in a virtual battlefield simulation) are updated by non-periodic *event updates*, which are relatively rare in terms of network traffic caused. These events will be communicated as they occur, regardless of the bandwidth and update rate settings. Changes in network conditions caused by non-periodic event transmissions are detected by the network estimation algorithm so that the operation of the framework is always adjusted according to the actual bandwidth used for transmitting periodic updates. Due to the scope of this article, this is not shown in our results.

On a final remark, the information model is trained in an off-line manner from pre-recorded trajectory on the local host and is transparent to the remote host. Thus, the framework is easy to be integrated in the current IM techniques. Information needed for notifying the switching decision to the remote host is negligible compared to the state update traffic and not considered in the following discussion.

In the next section, for the purpose to illustrate the proposed framework, the performance of the Information-Based Dynamic Extrapolation Model is evaluated through experiments run on two user motions that are representative of those entity motions for which higher-order extrapolations extrapolate more information, and there is the trade-off between "small-packet-fast-update" and "large-packet-slow-update" schemes as illustrated in Figure 1(b). The results presented demonstrate that the proposed technique can accurately choose the extrapolation model that gives the best information rate and consistency.

4. EVALUATION AND DISCUSSIONS

The Information-Based Dynamic Extrapolation Model is implemented on user motions from two representative NVE scenarios with simulated network conditions. For the purpose of this work to optimize extrapolation model selection based on the information metric, the network condition estimation is implemented merely as two control parameters, namely bandwidth and latency, instead of any particular estimation algorithm. This simple approach does not affect the applicability of the dynamic

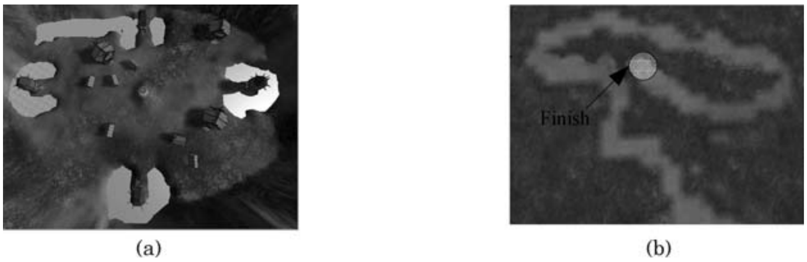


Fig. 3. Plain views of part of the environment for (a) the FPS game and (b) the racing game.

extrapolation model to a real Internet environment. The information model just needs the available bandwidth and latency to measure the information loss in transmitting data over a constrained network connection. The specific estimation algorithm, however, is independent of the operation of the dynamic extrapolation model and is therefore not discussed in detail in this article.

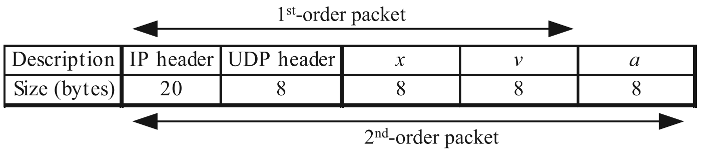
4.1 Experiment Set-Up

The two entity motion datasets used to evaluate the proposed framework are generated by a multiplayer First-Person Shooter (FPS) game [McCoy et al. 2007] and a racing game [Marshall et al. 2006] developed using the commercially available Torque Game Engine. The plain views of the two game environments are shown in Figure 3. In the FPS game (Figure 3(a)), an object can move around almost the whole space, except the water area denoted by blue areas. The goal of the players is to disable the opponent (programmed robots) by firing weapons. The disabled player is reborn at a random location and the game resumes. The motion is recorded at 10Hz for 1600s, and is then re-sampled at 50Hz by interpolating the recorded data using cubic spline interpolation. The re-sampling allows for evaluation of the proposed approach at higher update rates so that its efficacy to general NVEs could be shown through the results. We consider this data representative of those applications in which the simulation of realistic physics plays a key role, since the cubical interpolation provides natural continuity. In the racing game scenario (Figure 3(b)), a player can only run within the designed course denoted by the yellow lane, and the goal is to be the first player to reach the finish point. The racing motion is recorded in the game at 50Hz for 1600s. The two scenarios in Figure 3 are used to demonstrate that the differences in user behavior under various gaming scenarios and rules can be reflected in our information metrics. For illustrative purposes, the results presented are based on the x-coordinate $x\,(k)$ of the recorded motions.

The first half of the motions are used to build information assessments using (6) to measure the characteristics of the entity motions and the candidate extrapolation models. The second half of each motion is used as the test data to evaluate the dynamic extrapolation model. The training and test data each has a sample size of $N = 40000$, which is considerably larger than the number of unique entity states of 2226 and 1557 in the two games, respectively. The relative error ratio of the error term in (5) to the estimated mutual information is kept below 5% throughout the experiment. Given considerably larger amount of data, the information analysis can readily be applied to multidimensional data in a similar manner, but at the cost of increased computation burden. For the scenarios given in this work, the transition to a 2D dataset would require a sample size and computations about 3 orders of magnitude larger than the current settings. This seems bad, but the actual computation could be much lower than that because in an NVE environment the states that are visited by an object are typically sparse. The state space could be small. Also, computation cost can be reduced by using some advanced estimation methods mentioned in Section 2. In addition, the offline training process is

Table I. Network Condition Settings

Starting time (s)		800	880	960	1040	1120	1200	1280	1360	1440	1520
AB (kbps)		15	8	2	1	7	8	10	12	10	10
L_f (ms)		20	100	200	400	500	800	800	600	120	100
PPS	1^{st}-order	30	22.7	5.7	2.8	19.9	22.7	28.4	30	28.4	28.4
	2^{nd}-order	30	19.2	4.8	2.4	16.8	19.2	24	28.8	24	24
L_a (ms)	1^{st}-order	43	144	376	752	550	844	835	629	155	135
	2^{nd}-order	48	152	408	816	559	852	842	635	162	142



Description	IP header	UDP header	*x*	*v*	*a*
Size (bytes)	20	8	8	8	8

Fig. 4. Packet structure for the two extrapolation models.

conducted separately and thus the increased computation does not affect the run-time performance of the dynamic extrapolation.

The network conditions in the experiments are set up for a simplified but representative network environment for NVEs transporting synchronization messages using UDP. Typically, NVE participants are connected to each other via their residential broadband connection, also known as the "last mile link", and the core of the Internet, which has a much higher bandwidth. Therefore, the upstream link of the broadband connection, with the typical bandwidth value around 100kbps [Marshall et al. 2008], is the bottleneck link for this network. Considering the possibly multiple participants behind one last mile link due to the large number of users involved in an NVE and cross traffic from other applications such as internet video, the available bandwidth *AB* for a single NVE instance is set to be no more than 20kbps. The maximum update packet rate of the applications is 30 Packet Per Second (PPS). These settings are in line with general NVE traffic [Färber 2002; Feng et al. 2005; Harcsik et al. 2007].

As the dynamic extrapolation model operates by controlling the overall data rate to match the available bandwidth, there should be no queuing packet on the logical connection for the NVE instance on the bottleneck link. As shown in (10), the one-way network latency L_a from the local host to the remote host is modeled as the sum of packet transmission delay L_T on the bottleneck link and a delay L_f in passing the packets through the network nodes outside the last mile link. L_f includes all other factors in the overall latency except L_T, such as bit propagation delay subject to the speed of light and processing delay through hops in the core of the Internet. These elements in L_f are independent of the bottleneck bandwidth changes and are therefore collectively modeled as a fixed delay in the experiment

$$L_a = L_T + L_f = \frac{ps_i}{AB} + L_f. \tag{10}$$

The upper panel of Table I shows the changing network conditions given by the two control parameters *AB* and L_f in the test simulation, reflecting wide variation of the realistic Internet [Pantel and Wolf 2002a; Armitage 2003; Claypool 2005].

The rest of the experiment is set as follows. The set of candidate extrapolation models (as shown in (11)) consists of two widely employed models: $1st$ and $2nd$-order dead reckoning. The motion status parameters are entity position x, velocity v for the 1^{st}-order extrapolation and an additional acceleration a for the $2nd$-order extrapolation, each recorded in a data unit of 8 bytes. Considering the overhead in an UDP packet, the packet sizes for the two extrapolation models are respectively 44 bytes and 52 bytes (Figure 4). Such settings reflect the reality that periodic state updates are the dominating

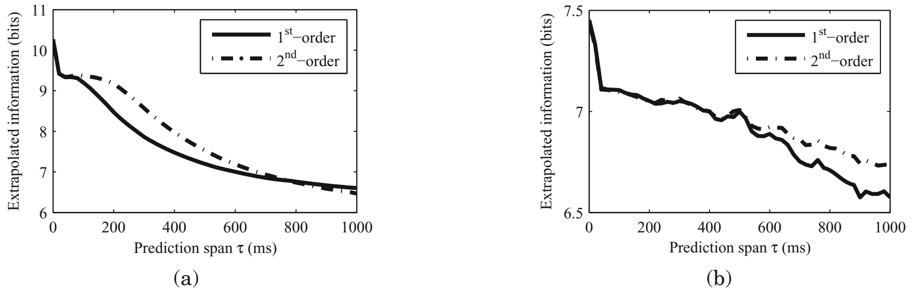


Fig. 5. Extrapolated information by the two candidate models for (a) the FPS motion and (b) the racing motion.

component of network traffic, and are representative of typical state updates in NVEs [Färber 2002; Feng et al. 2005; Harcsik et al. 2007]. The optimal update rates and the overall latencies under each of the network condition setting are summarized in the lower panel of Table I.

$$\begin{aligned} M &= \{m_1 = 1^{st}\text{-order DR}, m_2 = 2^{nd}\text{-order DR}\} \\ m_0 &= m_2 \\ U &= \{u_1 = [x, v], u_2 = [x, v, a]\} \\ PS &= \{ps_1 = 44 \text{ bytes}, ps_2 = 52 \text{ bytes}\}. \end{aligned} \tag{11}$$

The determination of the model switching time-out T_O mostly relies on the specific connection. In typical residence broadband connections, a short change that lasts for only seconds is regarded as transient [Marshall et al. 2008]. Therefore, T_O is set to be 5 seconds in our experiment. The convergence period T_C is arbitrarily set to be 2 seconds in this conceptual experiment, since there is no well-accepted threshold for user perception of model convergence.

4.2 Extrapolation Selection Based on the Information Measurement

The key component of the dynamic extrapolation model is the information model that serves as a performance measurement to pick one of the candidate prediction models delivering the highest information rate to the remote host under changing network conditions. The information model characterizes how much of the predictability of the true motion can be extrapolated from state derivatives by an extrapolation model.

Using (6), the information extrapolated by each of the two candidate models for increasing prediction spans is compared in Figure 5. The extrapolated information in bits measures the amount of knowledge that is used by the extrapolation model to predict an entity state with a certain level of certainty or confidence. Therefore, for a specific scenario, a higher extrapolated bit value indicates a higher probability that the predicted value is correct (i.e., the same as the true entity state). This is because that based on the information acquired from the latest update and the patterns in the user behavior, the possible value of the true state is narrowed down to a very limited set, which generally leads to very high fidelity.

For typical user motion from the FPS game (Figure 5(a)), an average amount of 10.2 bits data is needed to fully determine an entity state, among which over 9 bits can be extracted from the current $2nd$-order packet using standard $2nd$-order DR to estimate a future state at 200 ms later. Because of the information that is missing, the future state can never be determined completely by the extrapolation equation, and thus inconsistency arises from uncertainty. As expected, including the additional acceleration component does give the 2^{nd}-order extrapolation some advantage over the $1st$-order equation. The

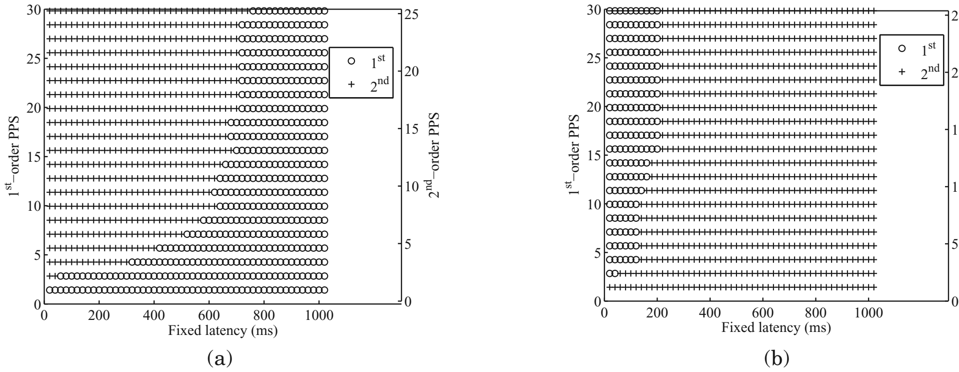


Fig. 6. Extrapolation choices based on information rate under varying network conditions for (a) the FPS motion and (b) the racing motion.

2*nd*-order extrapolation utilizes more information in most of the prediction spans presented, indicating that higher-order derivative provides a notable amount of knowledge about the entity states in the near future. However, the information extrapolated by the 2*nd*-order equation decays faster because of the sensitivity of the higher-order derivative to the rapid changes in motion status of the player, who moves in a wide territory chasing a moving target (Figure 3(a)). Therefore, the 1*st*-order extrapolation is generally expected to be preferred for predicting future states on longer time scales. Notice that the fact that the 2*nd*-order extrapolation utilizes more information than the 1*st*-order method over a extended range of prediction spans, which demonstrates that the interpolation has little effect on the overall motion since it only impact the data in time scale under 100 ms. The information metric results reflect the characteristics of human actions in the game.

From the measured extrapolated information in Figure 5(a), the motion in the racing game is generally smooth and stable since only less than 7.5 bits data could determine an entity state. In this scenario, players only move within the designed course and thus their motions are largely predictable. This feature is reflected by the information metric that 1*st*-order DR exhibits nearly equal predictability as 2*nd*-order DR for short prediction spans. In contrast to the FPS motion, 2*nd*-order DR exhibits information advantage over extended future period. In the racing game, with a static finishing point and the constrained path (Figure 3(b)), the entity motion would typically involve steady accelerations over extended periods. Notice that the extrapolated information converge for the two DR models during some prediction spans (e.g., between 800ms and 1000ms for the FPS game and 0ms to 500ms for the racing game). This indicates that the two extrapolations methods provide roughly the same amount of information. The additional data in the 2*nd*-order DR does not bring more information due to the patterns in the user behavior. The information metric shown in Figure 5 accurately captures features of the game motions, which eventually depend on the user behavior under specific scenarios and rules.

In spite of the disadvantage in utilizing information at specific time instances, the 1*st*-order extrapolation model may compensate for the average information rate by faster update generation. For any given available bandwidth and fixed latency, the overall information rate can be estimated from the information model, and the dynamic model switches the active extrapolation model to whichever produces the higher information rate. Figure 6 shows the selected active models for varying bandwidth and latency settings based on the trained information models for the two motions. To put the data rate setting in some perspective, bandwidth is presented by packet rate (PPS) instead of absolute bandwidth value. The difference of extrapolation choices between the two game scenarios comes from the information characteristics of the particular user motions (Figure 5). The model selection is determined by the extrapolated information within the functioning periods of the two models for a specific network condition. For the FPS motion, the 2*nd*-order extrapolation model is generally preferred in high-bandwidth and low-latency situations. The information advantage of the 2*nd*-order update message comes with the cost of a larger packet size. Unless the packets are transmitted to the remote host in time, the additional information and traffic load will be wasted. In low-bandwidth and high-latency scenarios, the 1*st*-order extrapolation is preferred because it provides simple but stable motion trends that outperform the sensitive higher-order derivative information in longer term prediction. For the racing motion, on the other hand, 2*nd*-order DR is mostly the favorable model due to its extended information advantage over 1*st*-order DR. However, a few exceptions can be found in situations with high bandwidth and low latency, where the functioning periods of the update messages reside in small prediction spans and the difference in information extrapolation between the two models is negligible. 1*st*-order DR is preferred for higher update rates. The contrast between the two game scenarios demonstrates that the information measurement is capable of capturing different types of user behavior and combining this factor with network conditions in making decision on selecting the best extrapolation model.

To evaluate the information model, the Information-Based Dynamic Extrapolation Model is carried out on the test data using the changing network settings listed in Table I. The varying available bandwidth and latency reflects variations in network conditions due to players that share the same last mile link joining in and leaving, or cross-traffic from other applications. For any given bandwidth, the overall data rates of the two extrapolations are tuned at the same level as the available bandwidth to send as much data as possible and in the meantime avoid queuing delay. It is obvious from Table I that the 2*nd*-order extrapolation always sends update packets at a rate about 20% lower than the 1*st*-order extrapolation because of the larger packet size, except the cases where the bandwidth is high enough to allow both the two extrapolations to update at the highest rate of 30 PPS. In addition, the 2*nd*-order extrapolation suffers from higher latencies than the linear extrapolation since it takes more time to transmit the larger packets through the bottleneck connection. It is then clear that the 2*nd*-order extrapolation faces negative impacts on both update rate and latency from the larger packet size. The overall effect of the two extrapolations under limited network resources depends on the tradeoff between packet size and update rate.

The implications of varying the active extrapolation model and packet rate on the remote information rate and inconsistency is shown in Figure 7 and Figure 8. First, for each of the 10 sections with different network settings, an active extrapolation model is selected based on the trained information model to deliver the higher information rate (Figure 6). The selected model is shown in Figure 7(a) and Figure 8(a) with numbered notations (① for the 1*st*-order extrapolation and ② for the 2*nd*-order extrapolation). Along with the average information rates rebuilt on the remote host by the two candidate models, the information rate of the dynamic model is also highlighted. The extrapolation model selection based on the trained information model (the numbered notations) is consistent with the actual information characteristics of the test data, and the dynamic model can accurately select the extrapolation model with higher information rate for different network conditions. Figure 7(b) and Figure 8(b) compare the remote inconsistency arising from using the two fixed extrapolation models and the dynamic model. Remote inconsistency is measured by simulation units, and averaged over every second. The interaction of the inconsistency that arises from employing either of the two fixed extrapolation models justifies the motivation of the dynamic extrapolation model because neither of the two fixed models minimizes remote inconsistency for all the network condition changes. Under changing network conditions, the 20% higher update rate of the first-order extrapolation could lead to contrary results in information rate and inconsistency compared to the second-order extrapolation.

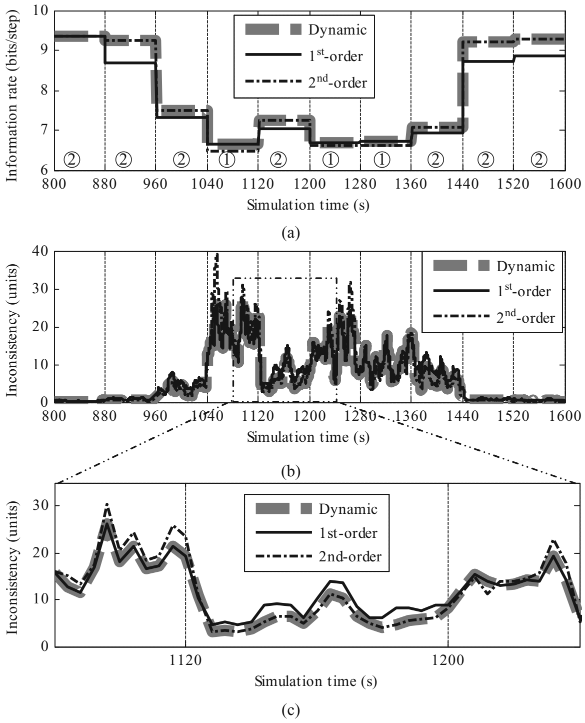


Fig. 7. Comparisons of (a) Average information rate (b) Remote inconsistency among the two fixed extrapolation models and the dynamic extrapolation model for the FPS motion. A highlighted section of (b) is shown in (c).

This also depends on the user behavior (evident contrast can be found between frames after 1040s and 1120s in Figure 7(b)). It is then evident that by dynamically switching to the extrapolation model that produces the higher remote information rate, the proposed technique minimizes remote inconsistency for varying network conditions. The dynamic extrapolation model accurately chooses the extrapolation model that produces lower inconsistency and thus outperforms statically employing either of the two fixed extrapolation schemes. In our results, a 0.2-bit increase in extrapolated information leads improvements in remote inconsistency at about 2-3 game state units. However, the improvement of a higher extrapolation information value depends on the specific application under consideration. For the purpose of clarity, model switchings over highlighted periods are shown in Figure 7(c) and Figure 8(c).

It is worth noting that in Figure 7(c), although the 1*st*-order extrapolation is the active model for the period starting from 1200 second, the 2*nd*-order extrapolation gives lower inconsistency at around 1210 second. The dynamic extrapolation model fails to pick this transient exception because the information-based model selection is the optimum in a statistical sense. It does not guarantee the best consistency at every time instance, but provides the most probable best model for long time implementation. Similar situations can also be found in Figure 8(b).

4.3 Discussion

The core of the proposed framework is the information model that relates predictability in user behavior and transmitted data, and hence combines both application and network layer elements in

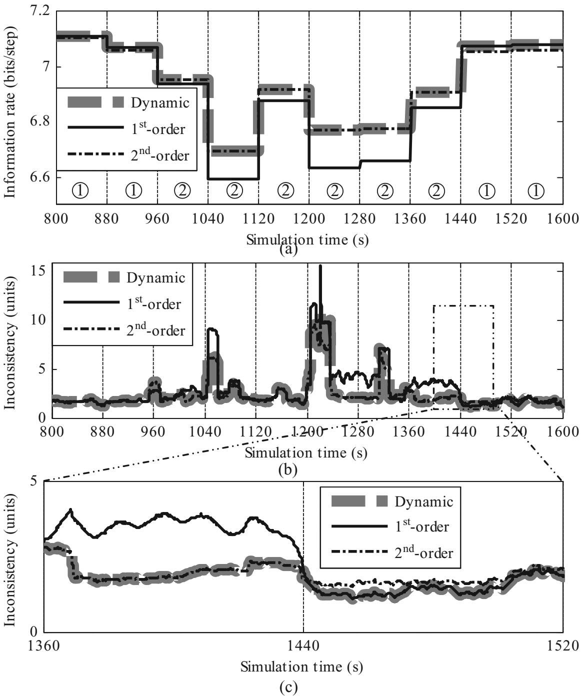


Fig. 8. Comparisons of (a) Average information rate (b) Remote inconsistency among the two fixed extrapolation models and the dynamic extrapolation model for the racing motion. A highlighted section of (b) is shown in (c).

adjusting system parameters. Under a certain bandwidth, there is one optimal update rate for each extrapolation (or ESU type) to transmit the highest information rate. Based on the information metrics, such an optimal update rate, together with latency, determines the information rate for each extrapolation model. The extrapolation method with the highest information rate is then chosen by the framework and the corresponding update rate is also employed. Such an approach guarantees that the usage of the available bandwidth is maximized for a given set of extrapolation models, and the highest information about the true entity motion is reconstructed by the selected extrapolation model that best suits the user behavior.

Using information and uncertainty instead of spatial or any other form of inconsistency provides a number of insights in examining and improving consistency maintenance. The information model first gives a novel perspective towards the trade-off between consistency and bandwidth usage: the synchronization messages are seen as providing partial information to reduce uncertainty about the remote entity state model; better consistency requires more information and thus update messages must be transmitted more often, which increases bandwidth usage. A quantified measure of this information perspective is provided by the information model. Second, the information metric of the synchronization messages takes into consideration both factors that affect consistency maintenance, namely the suitability of the extrapolation model and network conditions. If an extrapolation model gives poor remote consistency, the information measurements can help identify whether the problem is the extrapolation equation failing to utilize the predictability in the entity motion (a fast information decay) or a poor network connection that cannot support the necessary data transmission. In the former case, improving the underlying network connection alone would be of little help. One could either replace the state derivatives in the messages with some parameters containing more information about the entity state without changing the extrapolation algorithm [McCoy et al. 2007], or switch to a better algorithm that extrapolates the predictability into the remote state model. Finally, from the point of view of communication, a variable with lower uncertainty can be described and transmitted with less amount of data if properly encoded. The information metric of the user motion gives a quantified measurement of how much data is needed to rebuild the entity motion and how the level of rebuilt fidelity declines with time delay. Such an approach facilitates NVE deployment in that NVE designers can link desired application layer parameters, such as information rate, to network layer QoS requirements such as bandwidth and latency.

In our experimental studies, the active extrapolation model changes with network conditions. For the 1*st*-order extrapolation, the disadvantage of less information per packet can be compensated for by a faster update rate under some network conditions. The results presented here are representative of applications where more accurate physics are integrated for higher fidelity simulation (such as aircraft simulations and games that incorporates real world physics). In such applications, entity motion is simulated at a frequency higher than human perception, reaction and manipulation. So the entity dynamic is more affected by the physical law than human behaviors on very short time scales. These applications would thus have similar 2*nd*-order dynamics on the interpolation scales of interest to human perception, and the effects of the player manipulations probably only manifest themselves on longer time scales. For these applications, the 2*nd*-order extrapolation model would have an information advantage as shown in our experiment. However, it should be mentioned that including more data in the update packets does not always guarantee better consistency [McCoy et al. 2007], because whether or not a higher-order extrapolation can utilize the additional information in the messages depends on how well the model fits the entity motion. It is then fairly possible that for some application types a simple prediction method such as the linear model would utilize more information and therefore be always selected as the active model since it provides higher information rate without causing heavier traffic load. In such cases, employing the proposed dynamic framework, which would make a correct but static choice of the active model, seems unnecessary.

In the experiment results presented, the information model built from the training data exhibits good generalization on the test data. Using the information model built using the training data, the dynamic extrapolation model can precisely pick the active extrapolation model that gives higher information rate on the test motion. This provides a good indication that the information model successfully captures the dependence, or predictability, of the entity motion. Generally speaking, in order for the trained information model to be used in the dynamic extrapolation model, the application state dynamic is required to be stationary so that the training and test data may be considered statistically homogeneous. This is unlikely to be problematic for most NVEs, as sufficient user data can be collected over long time periods such that the limited spatial environment is well explored. Consequently the required data can be collected for accurate information calculation.

The consistency regulation in this work is based on network conditions of each individual user. Therefore, it is possible that the user using a poor connection is updated of the “true” situation later than others and so will be suffering from worse inconsistency. This is a consequence of how modern systems work. Heterogeneous clients making different extrapolation choices does not affect the quality (or user experience) and is more “fair” because every user experiences what is best based on its own connection. Taking a unanimous choice, on the other hand, would compromise the performance of the users with fast connections. In terms of fairness, our model does not make things any more unfair. In fact, it may make things more fair by not allowing congestion due to excessive traffic load. It is true that performance of the proposed framework as presented has assumed equitable router prioritization policies. However, extension of the work based on fairness as a result of more sophisticated router prioritization should not be a problem for interested adopters of the approach.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this work, the information management problem in NVEs is examined from an information theory perspective. By regarding update messages as an information carrier that delivers predictability of the future states to the remote host, the information model provides an analytical measurement of whether the data included in a message is worthy of the network traffic load it causes. Based on the information model, a dynamic extrapolation model is proposed to select the extrapolation model that can most efficiently make use of the available network bandwidth to reduce the uncertainty of the remote entity state model and optimize consistency. Results from experimental studies show that the Information-Based Dynamic Extrapolation Model can minimize inconsistency by making the maximal and most efficient use of the underlying network connection. Although only two polynomial extrapolation models are examined, the proposed dynamic extrapolation model is a general framework that can include any form of extrapolation algorithm.

Our future work shall focus on examination of the impact of jitter to the information model, and incorporating some practical bandwidth estimation so that the framework can be implemented for more realistic network conditions. More advanced extrapolation techniques could be included as part of the dynamic extrapolation model to achieve better consistency maintenance. Further improvement could be achieved as long as there is structure in the entity dynamics. For example, correlation of activities in multiple motion axis provides more information sources to aid entity state prediction. Although the framework is currently designed for regulating periodic updates and non-periodic events are simply sent immediately, it is worth noting that in case of a scenario where non-periodic events are not irregular (such as frequent weapon firing in a massive combat in military simulations). Then they become regular, which means they are very predictable and one need only transmit events corresponding to when the events are not triggered. The traffic for such “counter-events” would be thinner. Also, the idea of this work could also be generalized to any scenarios other than traditional movement (such as collaborative white board and media distribution) where approaches employed are predictive in nature. Although the exact procedure may not be directly applied to those scenarios, the basic philosophy to evaluate the efficiency of the transmitted data, that is, the consumed bandwidth, using information theory could be employed in general NVEs. In addition, the information characteristics of the entity motion is dependent on the specific user behavior of interest. To account for possible changes in user behavior during NVE deployment, an adaptive information model that can be trained in real-time using historical motion data within a sliding-window would be necessary. In such a case, the issue of the computational cost of the information calculation must be considered since, unlike the off-line training considered in this work, a computationally costly calculation may induce extra delay for state update in NVEs.

Aside from changing the amount of information inside the message, packets could also be dropped under constrained network bandwidth for a better consistency regulation. In fact, the regulation on periodic state updates can be seen as a special case of an optimistic game event synchronization scheme where a game server drops obsolete game events that lose their importance (i.e., executing them does not affect the final game state) by the time they arrive [Ferretti and Roccetti 2005]. Entity state update can be seen as a very simple case where the “event” to be considered is the latest position and motion status of the entity. Such information is always non-obsolete and must be delivered. The assessment of event obsoleteness is currently done by semantic analysis of the game, while information theory is also capable of measuring the correlation of the events in the evolution of the game. In other words, information metrics could aid the design of a novel selection scheme on game packet transmission.

REFERENCES

Armitage, G. 2003. An experimental estimation of latency sensitivity in multiplayer quake 3. In *Proceedings of the 11th IEEE International Conference on Networks (ICON'03)*. 137–141.

Calvin, J., Dickens, A., Gaines, B., Metzger, P., Miller, D., and Owen, D. 1993. The SIMNET virtual world architecture. In *Proceedings of the IEEE Annual International Symposium on Virtual Reality*. 450–455.

Capps, M., McGregor, D., Brutzman, D., and Zyda, M. 2000. NPSNET-V. A new beginning for dynamically extensible virtual environments. *IEEE Computer Graph. Appl. 20,* 5, 12–15.

Capps, M. and Stotts, D. 1997. Research issues in developing networked virtual realities: working group report on distributed system aspects of sharing a virtual reality. In *Proceedings of the 6th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 205–211.

Claypool, M. 2005. The effect of latency on user performance in real-time strategy games. *Comput. Netw. 49,* 1, 52–70.

Cover, T. M. and Thomas, J. A. 2006. *Elements of Information Theory.* 2nd Ed. John Wiley & Sons, Inc., Hoboken, NJ.

Delaney, D., Ward, T., and McLoone, S. 2003. On reducing entity state update packets in distributed interactive simulations using a hybrid model. In *Proceedings of the 21st IASTED International Multi-Conference on Applied Informatics*.

Delaney, D., Ward, T., and McLoone, S. 2006. On consistency and network latency in distributed interactive applications: A survey—part I. *Presence: Teleoperators Virtual Environ. 15,* 2, 218–234.

Färber, J. 2002. Network game traffic modelling. In *Proceedings of the 1st Workshop on Network and System Support for Games (NetGames'02)*. ACM, New York, NY, 53–57.

Feng, W.-c., Chang, F., Feng, W.-c., and Walpole, J. 2005. A traffic characterization of popular on-line games. *IEEE/ACM Trans. Netw. 13,* 3, 488–500.

Ferretti, S. and Roccetti, M. 2005. Fast delivery of game events with an optimistic synchronization mechanism in massive multiplayer online games. In *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE'05)*. ACM, New York, NY, 405–412.

Fraser, A. M. and Swinney, H. L. 1986. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A 33,* 2, 1134–1140.

Frécon, E. and Stenius, M. 1998. DIVE: A scaleable network architecture for distributed virtual environments. *Distrib. Syst. Eng. 5*, 91–100.

Hanawa, D. and Yonekura, T. 2006. A proposal of dead reckoning protocol in distributed virtual environment based on the taylor expansion. In *Proceedings of the International Conference on Cyberworlds (CW'06)*. IEEE Computer Society, Los Alamitos, CA, 107–114.

Harcsik, S., Petlund, A., Griwodz, C., and Halvorsen, P. 2007. Latency evaluation of networking mechanisms for game traffic. In *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames'07)*. ACM, New York, NY. 129–134.

IEEE. 1998. IEEE standard for distributed interactive simulation - application protocols. IEEE Std 1278.1a-1998.

IEEE. 2000. IEEE standard for modeling and simulation (M&S) high level architecture (HLA)—framework and rules. IEEE Std 1516-2000, i–22.

Kushner, D. 2002. The wizardry of id [video games]. *IEEE Spectrum, 39,* 8, 42–47.

Lee, B.-S., Cai, W., Turner, S. J., and Chen, L. 2000. Adaptive dead reckoning algorithms for distributed interactive simulation. *Int. J. Simul. Syst., Sci. Tech. 1,* 1–2, 21–34.

Marshall, D., McLoone, S., and Ward, T. 2008. Optimizing consistency by maximizing bandwidth usage in distributed interactive applications. *ACM Trans. Multimed. Comput. Commun. Appl. 6,* 4.

Marshall, D., McLoone, S., Ward, T., and Delaney, D. 2006. Does reducing packet transmission rates help to improve consistency within distributed interactive applications? In *Proceedings of the 9th International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games*. 88–92.

McCoy, A., Delaney, D., McLoone, S., and Ward, T. 2005. Dynamic hybrid strategy models for networked multiplayer games. In *Proceedings of the 19th European Conference on Modelling and Simulation (ECMS'05)*. 727–732.

McCoy, A., Ward, T., McLoone, S., and Delaney, D. 2007. Multistep-ahead neural-network predictors for network traffic reduction in distributed interactive applications. *ACM Trans. Model. Comput. Simul. 17,* 4, 16. 1276929.

McCoy, A. B. 2007. Data-driven modelling approaches for network traffic reduction in distributed interactive applications. Ph.D. thesis, National University of Ireland, Maynooth, Ireland.

Miller, D. C. and Thorpe, J. A. 1995. SIMNET: the advent of simulator networking. *Proc. IEEE 83,* 8, 1114–1123.

Moon, Y.-I., Rajagopalan, B., and Lall, U. 1995. Estimation of mutual information using kernel density estimators. *Phys. Rev. E 52,* 3, 2318–2321.

ORIGIN SYSTEMS. 1997. http://www.uoherald.com/news/.

PANTEL, L. AND WOLF, L. C. 2002a. On the impact of delay on real-time multiplayer games. In *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'02)*. ACM, New York, 23–29.

PANTEL, L. AND WOLF, L. C. 2002b. On the suitability of dead reckoning schemes for games. In *Proceedings of the 1st workshop on Network and system support for games (NetGames'02)*. ACM, New York, 79–84.

ROEHLE, B. 1997. Channeling the data flood. *IEEE Spectrum 34,* 3, 32–38.

ROULSTON, M. S. 1999. Estimating the errors on measured entropy and mutual information. *Physica D: Nonlinear Phenomena 125,* 3-4, 285–294.

SINGHAL, S. AND ZYDA, M. 1999. *Networked Virtual Environments: Design and Implementation* 1st Ed. Addison-Wesley, New York.

SINGHAL, S. K. 1996. Effective remote modelling in large-scale distributed simulation. Ph.D. dissertation, The Department of Computer Science, Stanford University, Stanford, CA.

STEUER, R., KURTHS, J., DAUB, C. O., WEISE, J., AND SELBIG, J. 2002. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics 18,* Supplement 2, S231–S240.

ZHANG, X., WARD, T., AND MCLOONE, S. 2008. Towards an information model of consistency maintenance in distributed interactive applications. *Int. J. Computer Games Tech. 2008,* 4, 1–10. 1453674.

ZHANG, X., WARD, T., AND MCLOONE, S. 2009. Exploring an information framework for consistency maintenance in distributed interactive applications. In *Proceedings of the 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT'09)*. 121–128.

Received June 2010; revised October 2010, January 2011, and April 2011; accepted April 2011