Hindawi Publishing Corporation
International Journal of Computer Games Technology
Volume 2008, Article ID 371872, 10 pages
doi:10.1155/2008/371872

Research Article

Towards an Information Model of Consistency Maintenance in Distributed Interactive Applications

Xin Zhang, Tomás E. Ward, and Séamus McLoone

Department of Electronic Engineering, National University of Ireland Maynooth, Maynooth Co. Kildare, Ireland

Correspondence should be addressed to Xin Zhang, xzhang@eeng.nuim.ie

Received 28 January 2008; Accepted 30 April 2008

Recommended by Jouni Smed

A novel framework to model and explore predictive contract mechanisms in distributed interactive applications (DIAs) using information theory is proposed. In our model, the entity state update scheme is modelled as an information generation, encoding, and reconstruction process. Such a perspective facilitates a quantitative measurement of state fidelity loss as a result of the distribution protocol. Results from an experimental study on a first-person shooter game are used to illustrate the utility of this measurement process. We contend that our proposed model is a starting point to reframe and analyse consistency maintenance in DIAs as a problem in distributed interactive media compression.

1. INTRODUCTION

Recent years have seen increasing interest in distributed interactive applications (DIAs). These are applications through which geographically distant end-users exchange information and interact with each other in a shared networked virtual environment. Historically DIAs first saw significant deployment in the military training realm (e.g., Simulation Networking system (SIMNET) [1], Distributed Interactive Simulation (DIS) [2]), and subsequently in the distributed virtual reality community (Naval Postgraduate School Networked (NPSNET) Virtual Environment [3]). However, in the last decade, it is the world of online entertainment systems that has seen the greatest proliferation of such technologies (e.g., Quake [4]). Despite the long evolution of this application class, one of the most persistent problems has been the issue of maintaining a uniform view of the simulation state for all users across the network, that is, the conditions of objects (or entities) and events in the shared environment, under the constraints of limited bandwidth and continuous user interaction.

Consistency refers to maintaining a spatially and temporally identical view of the data across the participating nodes (or hosts) in a DIA [5]. Due to the inevitable network latency which indicates the length of time taken in transmitting a message from one designated node to another [6], perfect consistency in DIAs is impossible to achieve [7] although the effects can be tempered through trading temporal fidelity for state consistency and vice versa [8]. This is known as the "*Consistency-Throughput Tradeoff*," which states that it is impossible for DIAs to have both a consistent and dynamical environment [9]. To deal with the tradeoff, various consistency maintenance mechanisms are employed to ensure a sufficient level of consistency. Generally, techniques used by these mechanisms can be classified into three classes [7]: information management techniques reduce the amount of data that has to be transmitted over the network; time management techniques manipulate time to mask the effect of network latency; system architecture techniques seek to improve the efficiency of processing and disseminating data. In this paper, we focus on one particular group of information management techniques, that is, predictive contract mechanisms that use prediction algorithms to reduce the number of update packets transmitted across the network. These mechanisms have been widely used in military training simulations and computer games [1, 10, 11].

Predictive contract mechanisms maintain controlled inconsistency, or a sufficient level of consistency, by using prediction schemes (e.g., Dead Reckoning [2, 5, 6], Hybrid Strategy Model [12], Nero-Reckoning [13, 14], etc.) to explore information about the future motion of the objects from contextual dynamics and reduce the frequency of sending entity state updates (ESUs) from the local controlling host to the remote host across the network. The ESUs are only generated and sent out to correct prediction errors larger than a given threshold on some inconsistency metric, while smaller prediction errors are simply ignored. Consequently, the local host only provides an approximated dynamic and encodes it in the ESUs. Predictive contract mechanisms sacrifice the accuracy of the remotely approximated dynamic in return for a reduction in the number of entity state updates and thus save bandwidth and reduce network latency. The performances of such consistency maintenance mechanisms depend mainly on how much the prediction model perceptually matches the real motion of the object on the local host.

Traditionally, performance of a prediction scheme is measured by the frequency of the ESU transmissions required to maintain the inconsistency within the threshold limit. The inconsistency caused by applying the prediction scheme in DIAs has been evaluated and analysed with different metrics of prediction error between states, and their approximations: drift distance (the average absolute error [15]), root mean square error (RMSE), and max norm (the worst error) [16] are all based on spatial difference between states of the same entity on different hosts; phase difference [17] considers temporal difference between the rendering time of the same entity state on different hosts; time-space inconsistency [18] takes into account both spatial distance and its duration. Unfortunately, none of the aforementioned measures gives explicit or quantified analysis of the contribution of the mechanism in helping reduce bandwidth consumption.

In this paper, we introduce a new framework which may aid in this regard and which utilises entropy and mutual information. Mutual information has been used in many other areas to detect and evaluate the dependence between different variables, such as gene expression [19], electrical signals from the brain [20], and so forth. In our model, mutual information is employed to measure the dependence between the real state dynamic and the approximated state dynamic on the local host. The inconsistency induced by discarding prediction errors within the threshold limit is measured as the information loss in the local approximation, which also indicates the theoretical bandwidth saving because mutual information is a direct and quantified measure of the minimal amount of data required to fully describe the interdependence between two variables, namely, the real and approximated motion in the context of DIA. The performance metric of the prediction algorithm is its ability to make use of ESUs to explore information about entity motion and reduce the amount of data required to maintain consistency. By investigating the use of information theory as a measurement of both components, we are able to provide a quantified model to analytically study the "*Consistency-Throughput Tradeoff*" as a problem in lossy source coding on the local host, that is, how good is the local host in providing entity dynamic information and what is the cost for that quality of sharing object motion.

In our model, the complexity of the entity motion is measured by entropy which is calculated from sampled probabilities. There are other advanced and more accurate approaches, such as fuzzy logic and neural networks, used to model object behaviour. An overview of these techniques can be found in [21]. The simple probability model we used here captures the underlying assumption behind all these behaviour models; that users will act similarly under similar circumstances, which makes our approach applicable to any extrapolation method. It is also worth noting that mutual information provides a general measure of the interdependence between the locally generated states for a given entity and those simulated to be rendered remotely. Other statistical measures are also available for measuring the dependence between the real and approximated dynamics. However, most of them only measure specific dependence patterns, such as linearity in the case of the Pearson's correlation. In DIAs, where the motion of the object is usually nonlinear and complicated, such dependence metrics could be misleading [22]. It should be stated that regardless of the measure used, the work reported here is the first such attempt to use any measure of this dependence as a richer measure of compression.

The remainder of this paper is organised as follows. A mathematical background of concepts and methods in information theory that are employed in our model is given in the next section. This is followed by fundamental principles of predictive contract mechanisms and detailed explanations of our information model to formulate local information processing in Section 3. The description of the experimentation is given in Section 4, while Section 5 presents our results and discussion. Finally, the paper ends with conclusions and directions of future work in Section 6.

2. BACKGROUND

We begin with a brief review of the concepts of Shannon entropy and mutual information in information theory and introduce the numerical procedures to estimate them from experimental data. All the definitions and methods here are given in discrete terms, as states of all variables in a virtual environment, however vivid, are finite and discrete.

Consider a random variable X with m possible states $\{x_1, \ldots, x_m\}$, each with probability $p(x_i)$. The entropy $H(X)$ of the variable is defined as [23]

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

Entropy measures the degree of complexity of variable X. In the completely determinant case, some state x^* is such that $p(x^*) = 1$, and all other probabilities are zero, we have $H(X) = 0$. If, on the other hand, there is a universal probability $p(x_i) = 1/m$ for all possible states, the maximal entropy is $H(X) = \log m$. In general, variables with larger entropy are more complex and more unpredictable. From the view of compression, entropy also indicates the minimal length of data required to fully describe the variable.

For two random variables X and Y, the remaining complexity of X given knowledge about Y is defined as conditional entropy:

$$H(X \mid Y) = \sum_{x_i, y_j} p(x_i, y_j) p(x_i \mid y_j), \tag{2}$$

where $p(x_i, y_j)$ denotes the joint probability that X is in the state x_i and Y is in the state y_j, and $p(x_i \mid y_j)$ denotes the conditional probability that X is in the state x_i, given Y is in the state y_j. For arbitrary variables, entropy is larger than conditional entropy. The difference between (1) and (2) is the amount of reduced complexity of X from knowing information about Y. Thus, the mutual information $I(X;Y)$ is defined as [23]

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

Mutual information measures the interdependence between the variables X and Y. Notice that entropy is the automutual information between the variable X and itself, that is, $I(X;X) = H(X)$. The mutual information in (3) is also called cross mutual information [20].

All the concepts mentioned here involve knowledge about respective probability functions, which are normally not known in practice. For the purpose of this paper, we use a simple approach to estimate probability and mutual information from experimental data, though other more advanced and complicated algorithms exist [19, 24].

Consider a sequence $x(t)$, $t = 1, 2, \ldots, N$ as a collection of N samples of X at different time instances t. Let r_i be the number of cases that $x(t) = x_i$. The probabilities are estimated as the frequencies of occurrences [19]:

$$\widehat{p}(x_i) = \frac{r_i}{N}. \tag{4}$$

Similarly, the joint probability of two variables X and Y can be estimated from two sample sequences $x(t)$ and $y(t)$ with the same length according to

$$\widehat{p}(x_i, y_j) = \frac{r_{ij}}{N}, \tag{5}$$

where r_{ij} is the number of cases that $x(t) = x_i$ and $y(t) = y_j$ at the same time. With these probabilities calculated, we can estimate mutual information between X and Y as defined in (3). However, it is known that the estimation of mutual information from limited-length samples is systematically biased due to the finite size effect. The systematic error can be corrected by applying an additional term to the original definition [19, 25]:

$$\begin{aligned} I(X;Y) \approx &\sum_{x_i, y_j} \widehat{p}(x_i, y_j) \log \frac{\widehat{p}(x_i, y_j)}{\widehat{p}(x_i) \widehat{p}(y_j)} \\ &- \frac{m_{xy} - m_x - m_y + 1}{2N}. \end{aligned} \tag{6}$$

Here, m_x, m_y, and m_{xy} denote the number of different state combinations with nonzero probability. The sample size must be considerably larger than the number of possible state combinations to make a good estimation.

The definitions and estimations presented above allow us to model and analyse predictive contract mechanisms in our new framework, in which maintaining controlled inconsistency is viewed as information sharing with loss. We will use entropy and mutual information to quantify the local information generation and processing in the next section.

3. INFORMATION MODEL

As described before, DIAs use predictive contract mechanisms to reduce data transmission requirements. One of the most common techniques employs a concept called Dead Reckoning [2, 5, 6] to extrapolate state from ESUs. The IEEE DIS standard in particular advocates such methods and further classifies the predictive contract mechanisms into two main components: prediction and convergence [2, 9]. In this article, we only focus on the prediction and reconstruction operated locally. The convergence algorithms as well as network latency, which will certainly affect remote inconsistency, are not included in our current model. Nevertheless, for the convenience of the reader, we will briefly mention the convergence algorithms to give a complete picture of the underlying principles behind predictive contract mechanisms.

The prediction algorithms are the core of the whole mechanism, because they define how the actual entity states are locally packed, with loss, into ESUs, and then reconstructed. In standard Dead Reckoning and its various extensions, multiple-order polynomial functions are used to extrapolate state evolution until the next ESU is generated [2, 26, 27]. More complicated methods involving statistical learning, such as Kalman filters [28] and Neural Networks [13, 14], are employed to improve the performance of prediction. Whether having a closed form formula or not, these algorithms are essentially functions or mappings $f(\cdot)$ from the previous generated ESUs to the anticipated states in the future.

The convergence algorithms define how entity states on remote hosts are corrected, on receiving an ESU, from the inaccurate estimation to its real value, so that the approximated dynamics look more natural and smooth. Currently, polynomial equations are the most commonly used convergence algorithms [9, 26, 29]. Higher-order equations generally generate smoother converging trajectory than low-order equations but they require more computation. The convergence operation is taken after the arrival of an ESU on the remote host to gain better visual perceptual consistency [9] and is thus not considered by the local host in issuing ESUs.

As shown in Figure 1, at each simulation time-step, the local host checks the error between the extrapolated state and the actual entity state. The predicted value is accepted, and the simulation goes on if the error does not exceed the threshold; otherwise an ESU including data required by the prediction algorithm is generated to be sent. Most systems

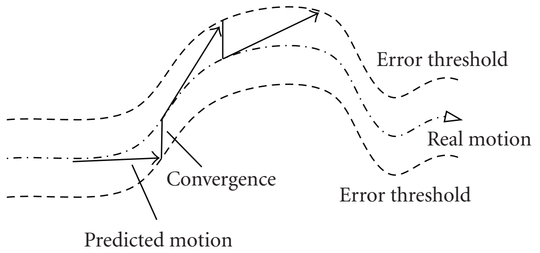


Figure 1: Visual illustration of Dead Reckoning procedures. In this case, we use linear extrapolation and zero-order (or snap) convergence.

send out an update if there is no ESU sent within a timeout period just to inform the remote host that the object is still "alive." These updates are relatively rare compared to the regular ESUs and are not related to local extrapolation, thus they are not considered in this paper. To reconstruct the approximated dynamic remotely, the remote host employs the same prediction model to regenerate the extrapolated states and applies the convergence algorithm on receiving ESUs.

The diagram in Figure 2, with the notation shown, presents our information model in which we seek to reframe the local operations of predictive contract mechanisms as information generation, encoding, and reconstruction processes. The basic idea is that the simulation cycle or "game loop" in a gaming systems context is generating information, that is, it generates updates to entity states at a rate suitable for high fidelity rendering for the local user. By using prediction models and thresholds, predictive contract mechanisms prune the generated information and encode the remainder into ESUs. As such, only an approximated dynamic of the entity is provided by the local host. Less data is required to transmit the approximated dynamic because part of the information is discarded, and the bandwidth saving should equate to the amount of information loss.

3.1. *Information generation*

As mentioned previously, the shared virtual environment is spatially and temporally discrete. Therefore, measurements of the entity state over time yield the discrete time series $d(k) = \{d(1), d(2), \ldots\}$, where k is the index of simulation time step and the value of $d(k)$ varies within a finite discrete set of entity state values $S = \{s_i\}$. By rendering the entity state at each time step, the local host is generating information about the state. The amount that is dependent on the complexity of the entity's motion can be characterised by a probability function $p_d(s_i)$. The average amount of information generated by the local host at each time step is the amount of uncertainty of the motion, that is, the entropy $H(d)$:

$$H(d) = -\sum_{s_i} p_d(s_i) \log p_d(s_i). \qquad (7)$$

In the case of a static environment where the entity state remains at one particular value all along (there is some state s^* such that $p_d(s^*) = 1$), the entropy of this motion would be zero, meaning that there is no information to be shared, and thus no ESU is needed and the remote view of the environment will be consistent with that on the local host. Entity motions with larger entropy require more information in order to fully replicate the state evolution.

3.2. *Information encoding*

Predictive contract mechanisms reduce the network traffic required for the DIA at the cost of losing state fidelity on the remote host: the local host only provides a pruned dynamic that resembles the real one at some level. This includes two different approximations: ESUs are sent out at a lower frequency than the simulation cycle; each update only contains partial information about the entity motion. Therefore, only part of the information generated on the local host is embedded in ESUs to be sent to the remote host, and the rest is discarded. The tolerable loss of fidelity is controlled by an error threshold. This can be seen as a lossy source coding or lossy media compression.

Let the time series $\hat{d}(k) = \{\hat{d}(1), \hat{d}(2), \ldots\}$ denote the approximated entity dynamic simulated by the local host using predictive contract mechanisms. We use cross mutual information $I(d; \hat{d})$ to measure the amount of information successfully delivered from the real dynamic to the approximated one. The information loss $\mathrm{IL}(d; \hat{d})$ in the process of estimating d as $\hat{d}$ (which is also the reduced bandwidth requirement due to a nonzero threshold) is the remaining uncertainty of the real motion d after we have the approximated $\hat{d}$:

$$\mathrm{IL}(d; \hat{d}) = H(d \mid \hat{d}) = H(d) - I(d; \hat{d}). \qquad (8)$$

3.3. *Local information reconstruction*

To reconstruct the dynamic, the prediction algorithm extracts information about d embedded in the ESUs and interprets it in the form of the approximated state dynamic afterwards. We use time-shift cross mutual information [20] to measure the amount of information utilised by the prediction model. Cross mutual information between any two time series $x(k)$ and $y(k)$ with a time shift τ is defined as [20]

$$\begin{aligned} I(x; y_\tau) &= I(x(k); y(k+\tau)) \\ &= \sum_{x(k), y(k+\tau)} p_{xy}(x(k), y(k+\tau)) \log \frac{p_{xy}(x(k), y(k+\tau))}{p_x(x(k))\, p_y(y(k+\tau))}. \end{aligned} \qquad (9)$$

Here, the time shift (or delay) τ refers to the difference between the indices of the time-steps in the two sequences. $I(x; y_\tau)$ is the average amount of information contained in the sequence $x(k)$ that can be learned about $y(k)$ at τ steps later. If the local host is generating an ESU at each time step, we will have the time series $u(k) = \{u(1), u(2), \ldots\}$

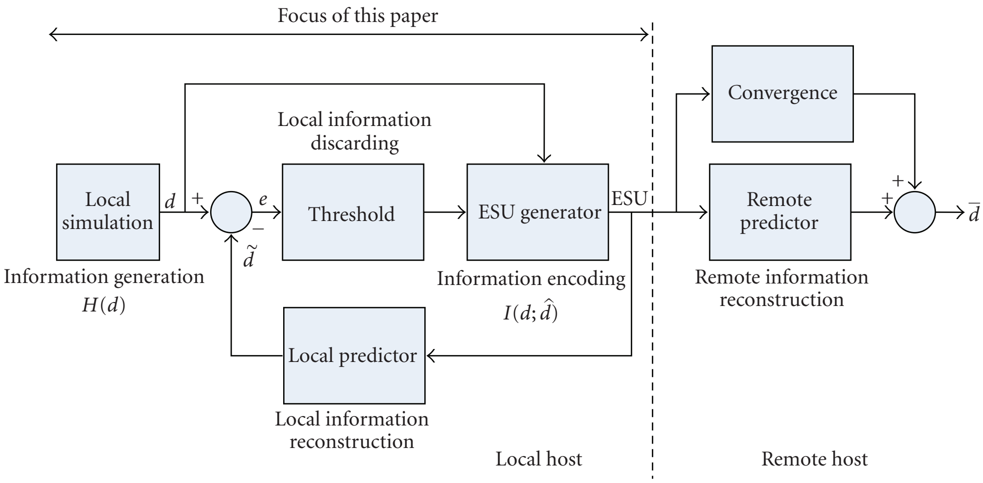


Figure 2: Information model of predictive contract mechanisms. Information generated by local dynamic d is encoded with loss into the ESUs, from which only an approximated dynamic $\hat{d}$ can be reconstructed. The extrapolated dynamic $\tilde{d}$ is compared to the real dynamic to decide whether an ESU is needed to correct the prediction error. $\bar{d}$ is the approximated dynamic reconstructed remotely by the remote predictor and the convergence algorithm.

representing the value of the potential ESU at time-step k. According to (9), the term $I(u;\hat{d}_\tau)$ is the average information that is contained in the ESU sent at time-step k and is used to predict the state at time-step $k+\tau$. Therefore the total information in a single ESU employed by the prediction model should be the sum of all the time-delayed cross mutual information $I(u;\hat{d}_\tau)$, as long as the ESU $u(k)$ is referenced in predicting $\hat{d}(k+\tau)$. For example, in standard Dead Reckoning, only the latest received ESU is used to predict states until the next ESU is generated, so the average information acquired by the prediction model from that ESU is

$$I_U = \sum_{\tau=0}^{l-1} I(u;\hat{d}_\tau), \tag{10}$$

where l is the average time interval between two successive ESUs. l also indicates the number of time-steps that an ESU is employed in prediction, that is, from the time this ESU is generated until the time before the next ESU generation. Thus, (10) calculates the effective information delivered by a single ESU to the local approximation.

With the ESUs being the only source of information to reconstruct entity dynamics, we have

$$I(d;\hat{d}) = \frac{n}{N}\cdot I_U, \tag{11}$$

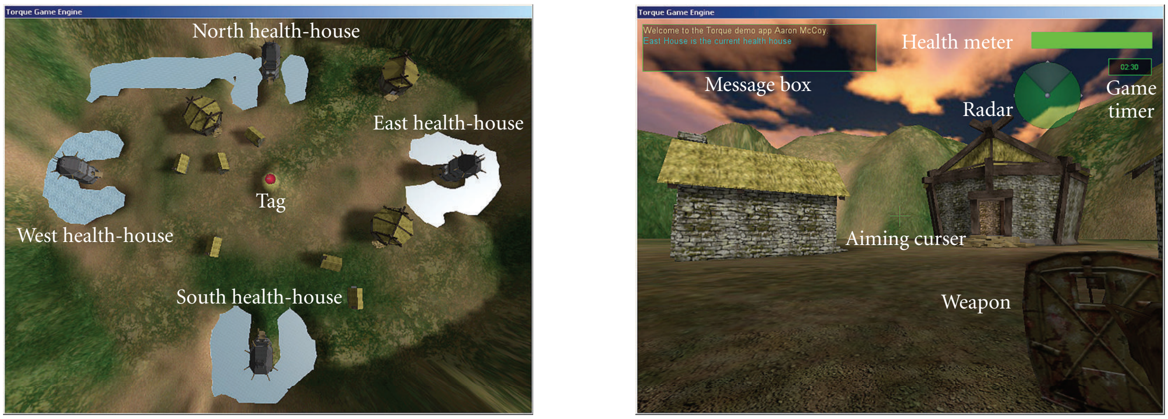
where n is the number of ESUs during N simulation steps.

Equations (8) and (11) express the core of our information model that information about entity state evolution over time, generated by local hosts, is encoded in ESUs and can be regenerated with some loss by employing prediction algorithms. Actually, (10) and (11) imply an "*Accuracy-Computation tradeoff*" between prediction accuracy against computational and memory resource overhead [9, 26], because computation also takes time and compromises consistency. Simpler models like standard Dead Reckoning only require a single ESU to extrapolate entity states; more complicated methods improve prediction accuracy and further reduce bandwidth consumption, at the cost of additional memory (by referencing longer historical records and more ESUs) and computational resources.

With the model and general procedure described above, we are able to measure how information is generated and processed locally to reduce the amount of data transmitted to maintain consistency, and thus conserve bandwidth. Although the calculations are demonstrated with one-dimensional time series, our approaches can be extended to higher-dimensional movements by using joint probability and joint mutual information [23]. Consequently, calculations in higher-dimensional data would require larger sample sizes to vanish the finite-size effect as (6) states. In the next section, we apply this framework to a multiuser first-person shooter game. The game settings are representative of computer game interactions. The results show that our model is applicable to general predictive contract mechanisms.

4. EXPERIMENTAL DATA

The practical game scenario we use here to show how the proposed framework works is a multiplayer first-person shooter (FPS) game developed using the commercially available Torque Game Engine [26]. The game scenario is shown in Figure 3. The goal of the players in the battlefield is to hold the special "tag" item; in the meantime they can attack each other with their weapons. Players can replenish their health meter in the health-houses. The last survivor holding the "tag" wins. This "deathmatch" scenario is fairly typical in online FPS games. For the convenience of illustration, our numerical study is based only on the x-coordinate of the user. Therefore, the motion of the entity is a one-dimensional dynamic and is recorded as a scalar time sequence $d(k) = \{d(1), d(2), \ldots\}$.



(a)

(b)

Figure 3: The FPS game scenario.

The predictive contract mechanism we examine here is Dead Reckoning (DR). We consider both linear and second-order extrapolations. The inconsistency threshold used here is the spatial distance metric. Let $\hat{d}(k) = \{\hat{d}(1), \hat{d}(2), \ldots\}$ be the estimated dynamic. At each simulation time-step k, the linear extrapolation for the current entity state is

$$\tilde{d}_1(k) = d(k_u) + (k - k_u)v, \tag{12}$$

where v is the estimated velocity in the latest ESU generated at time-step k_u. The extrapolated value $\tilde{d}_1(k)$ is accepted if the prediction error does not exceed the given threshold h; otherwise an ESU containing the current state value and velocity estimation is generated based on the real dynamic, that is,

$$u_1(k) = \{d(k), v = d(k) - d(k-1)\}. \tag{13}$$

Here, state $d(k)$ replaces $\tilde{d}_1(k)$ as a correction. For second-order DR, extrapolated entity state and data in an ESU are given by (14) and (15), respectively,

$$\tilde{d}_2(k) = d(k_u) + (k - k_u)v + \frac{1}{2}a(k - k_u)^2, \tag{14}$$

$$\begin{aligned} u_2(k) = \{&d(k), v = d(k) - d(k-1), \\ &a = d(k) + d(k-2) - 2d(k-1)\}, \end{aligned} \tag{15}$$

where a is the estimated acceleration.

In our experiment, two players are asked to play against each other. Our experiments were conducted for the simulation interval $T_s = 100\,\text{ms}$ and varying error thresholds h. On obtaining both the real and approximated dynamics, we examine the information processing as stated in the previous section. Results and discussion are presented in the next section.

5. RESULTS AND DISCUSSION

Figure 4 shows the real and the estimated dynamics for two different thresholds. As expected, the larger threshold leads to less dynamical approximated entity states, which contains less information about the real dynamic. From state trajectories, it is clear that the two extrapolation methods used here make little difference in approximating entity motion when the threshold is small. The second-order extrapolation causes more intense oscillations when the threshold is large, since more previous states are referenced in order to estimate the acceleration. The effect on prediction errors takes longer time to vanish in "high jerk" motions such as our FPS game.

Figure 5 shows the estimated probability function of the entity states. The probability is unevenly distributed among all possible states. States with significantly higher probability are the places where the player holds its position seeking targets (either the "tag" or the opponent), while those unlikely states are positions the player passes while chasing. The motion of the object generates more information in chasing moments than waiting periods.

Equations (4) and (7) give an entropy $H(d) = 9.2406$ bits. This is the average length of data required per simulation step to fully describe the dynamic. Some of this information generated by the local entity dynamic is lost because errors less than the threshold are simply ignored. The larger the threshold is, the more information is discarded and thus less ESUs are needed to deliver it. In Figure 6, we present information loss and number of ESUs for varying thresholds, along with three different traditional inconsistency metrics:

$$\begin{aligned} \text{drift distance} &= \frac{1}{N}\sum_k |d(k) - \hat{d}(k)|, \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum\nolimits_k (d(k) - \hat{d}(k))^2}, \\ \text{max norm} &= \max_k |d(k) - \hat{d}(k)|. \end{aligned} \tag{16}$$

In Figure 6, all measurements are normalised, and information loss is presented as a percentage of $H(d)$. It can be seen that for both extrapolation methods, information loss agrees with traditional metrics, following similar trends. The

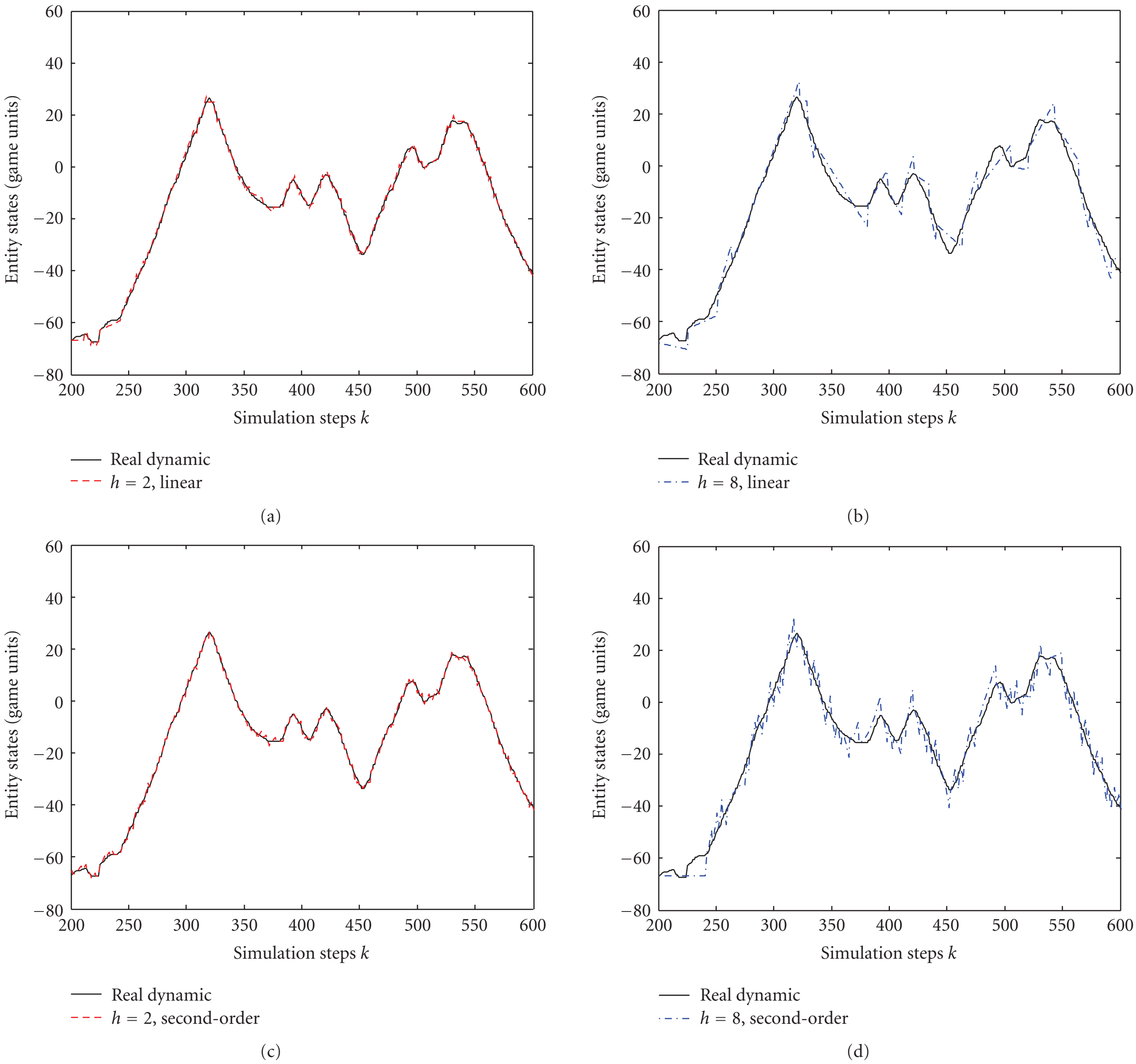


FIGURE 4: The real and estimated dynamics for (a) threshold h at 2 using linear DR extrapolation, (b) threshold h at 8 using linear DR extrapolation, (c) threshold h at 2 using second-order DR extrapolation, and (d) threshold h at 8 using second-order DR extrapolation.

advantage of our information loss measure over the others is that it not only measures to what degree the approximated dynamic resembles the real one but also indicates the bandwidth saved through tolerating inconsistency. In this example, the second-order extrapolation delivers more information to the approximated dynamic than the linear extrapolation.

Figure 6 is also an illustration of the "*Consistency-Throughput Tradeoff*" and provides guidance for a designer to pick a reasonable threshold for a given available bandwidth. For example, with the current prediction algorithms and game scenario, Figure 6 suggests an optimal threshold h between 10 and 20 saves over 80% ESU transmission while losing only around 20% of the information. Larger thresholds lead to little further reduction in ESU transmission, while information loss increases significantly (especially for linear extrapolation); and further reducing information loss would cause significant increase in network traffic.

Another issue worth mentioning is that information loss increases rapidly as threshold increases. As stated before, information loss is the discarded information about entity state evolution and also measures the reduced bandwidth requirement per time-step due to a nonzero threshold. Therefore, we could expect the same reduction rate in the number of ESUs, which indicates the practical bandwidth consumption. However, in Figure 6(a), we can see that frequency of ESU transmission decreases much slower. This observation suggests that ESU transmission can be

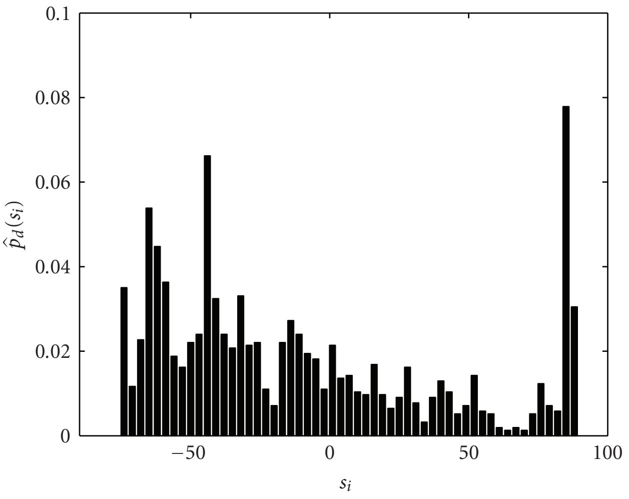
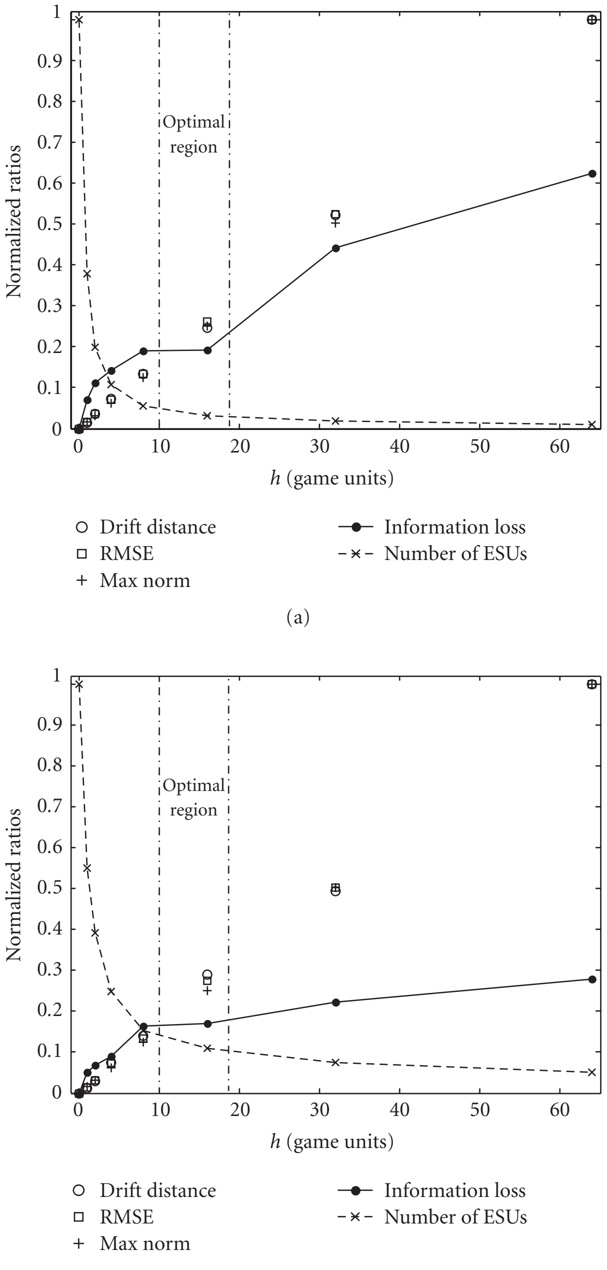


FIGURE 5: The estimated probability function of the entity states. Possible entity state values s_i vary from −80 to 80.



(a)

(b)

FIGURE 6: Normalized information loss, three different traditional inconsistency metrics, and number of ESUs for varying thresholds. Results are presented in terms of normalised percentages for (a) linear DR extrapolation and (b) second-order DR extrapolation.

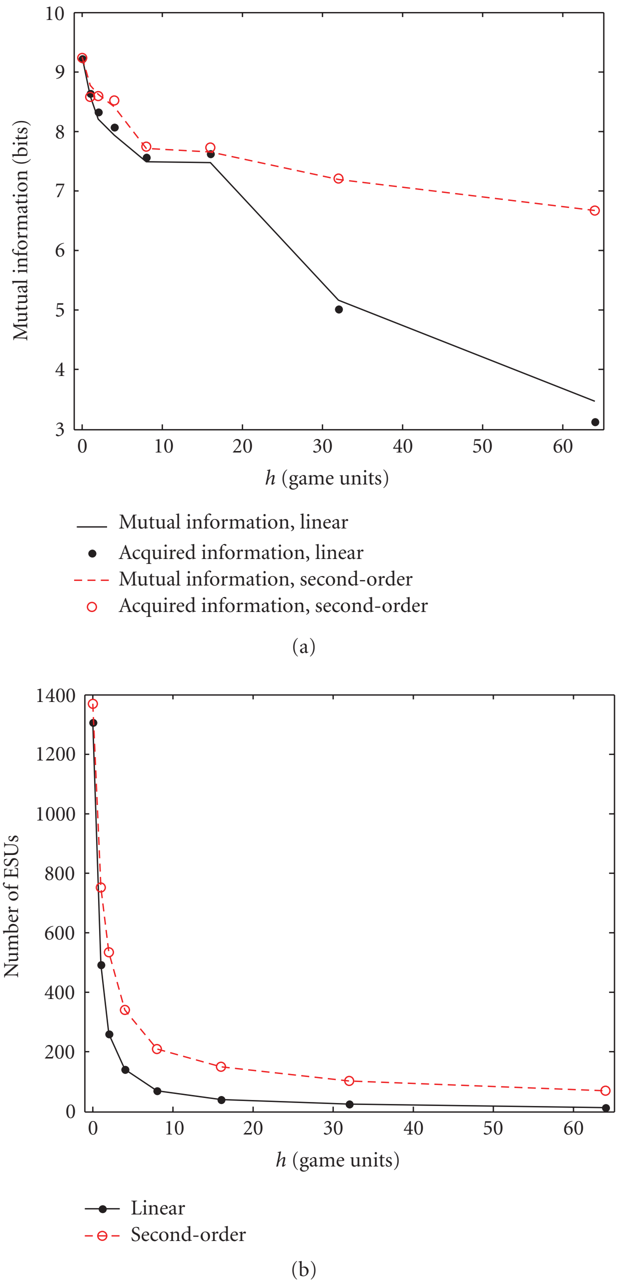
further reduced by applying external compression algorithms: encode them as a signal sequence, and decode them on the remote host. So far, the protocol independent compression algorithm (PICA) has been used to reduce the bit rate in DIAs [30]. It operates by sending only the byte difference between the current ESU packet and a reference packet. But this algorithm does not consider any statistical aspects of the ESUs and is not optimal. The bandwidth used to transmit the ESUs can be further reduced to the theoretical boundary implied by our formulations if statistical compression methods are employed.

We also examine (11) which is how the prediction model acquires information from the ESUs (see Figure 7(a)). Here, the average time-steps l between two ESUs are estimated by

$$l = \frac{N}{n}. \tag{17}$$

Numerical results presented in Figure 7(a) confirm (11) and that information encoded in ESUs is utilised by the prediction model in reconstructing the approximated dynamic. Here, prediction algorithms do not generate or store any information about entity states. Their contribution in saving data transmission is that they interpret information contained in the ESUs, and explore the future entity states from previous motions, and thus reduce redundant update packets. More advanced prediction algorithms match entity dynamics better and can extract more information from the ESUs. This superior performance generally comes from two factors: larger network traffic and more computational resource requirements. For instance, in Figure 7(a), the second-order extrapolation is extracting more information per time step from the latest ESU than the linear extrapolation, at the cost of a larger number of ESU transmissions (see Figure 7(b)) and more memory to restore longer referenced historical states and more computational resources to calculate the extrapolation from (14). Even though computational consumption is becoming more and more insignificant as computers are becoming more powerful, our model still provides a formulation of the ability of the prediction model to utilise ESUs by (10) and an explicit way to deal with the tradeoff between prediction accuracy and computational consumption.

All the formulations in our information model are based on entropy and mutual information between the real and approximated dynamics and are independent of the prediction algorithms applied. Hence, the proposed model is applicable to general predictive contract mechanisms.



(a)

(b)

Figure 7: (a) Mutual information between the local and remote dynamics, and total information acquired from ESUs in bits per time-step. (b) Number of ESUs for the two extrapolations.

Notice that our information model of predictive contract mechanisms resembles some media compression algorithms. For example, medical images are compressed by encoding quantised prediction errors using differential pulse code modulation (DPCM) [31]. The video compression algorithm MPEG follows similar procedures by employing motion compensation to estimate position of objects in the next frame, and only transmitting the difference transformed by discrete cosine transform (DCT) [32]. Information loss in these media compressions is induced by quantisation. Similarly, in predictive contract mechanisms any prediction error less than threshold is quantised to zero and then discarded. The role of the predictor is to eliminate redundant information between frames in media, or entity states in DIAs, because we can learn some information about future states by exploring previous data. Therefore, it should be pointed out that our framework is a starting point to view consistency maintenance in DIAs as a problem in media compression. Using information theory, we can employ methods in media compression to improve current state-of-the-art communication protocols in DIAs.

6. CONCLUSIONS AND FUTURE WORK

This paper has shown how, by employing information theory, entity state evolution can be viewed as an information generation process, and how predictive contract mechanisms can be modelled as a lossy information compression and reconstruction process. Analytical results show that the local host reduces the amount of data required to maintain some level of consistency by discarding part of the information generated by local entity state evolution. The remaining information is encoded in the ESUs and can be utilised by a prediction model to reconstruct a simpler and less dynamical approximation to the actual entity states.

Through numerical studies, our mutual information metric agrees with traditional inconsistency metrics. Moreover, the advantage of mutual information is that it not only can be seen as an inconsistency metric but also provides the theoretical bandwidth saving which can be achieved by applying prediction models. Our results also suggest that the bit rate in DIAs can be further reduced by applying external compression algorithms that consider the statistical aspects of the ESU sequence. Procedures presented here could shed light on designing optimal prediction algorithms to deal with the "*Consistency-Throughput Tradeoff*" and the "*Accuracy-Computation tradeoff*." Employing information formulations, our model to reframe consistency maintenance as distributed media compression is a novel and promising philosophy in the study of DIAs.

The model as developed in this paper has considered only the information processing on the local host. Within this information-theoretic framework, this is viewed as a form of lossy source compression. The picture of course is far from complete, and in subsequent work we will extend the analysis of predictive contract mechanisms to include the effects of the channel and the decoding or reconstruction of the original state. In particular, we believe that non-ideal attributes of the communication channel such as latency, packet loss and finite bandwidth have insightful interpretation within our framework.

ACKNOWLEDGMENTS

This work is supported by the Irish Research Council for Science, Engineering, and Technology (IRCSET): funded by the National Development Plan. The authors would like to thank Dr. Aaron McCoy for providing the user motion data used in the experimental study.

REFERENCES

[1] J. Calvin, A. Dickens, B. Gaines, P. Metzger, D. Miller, and D. Owen, "The SIMNET virtual world architecture," in *Proceedings of IEEE Virtual Reality International Symposium*, pp. 450–455, Seattle, Wash, USA, September 1993.

[2] IEEE Standard, "IEEE standard for distributed interactive simulation—application protocols," *IEEE Std 1278.1a-1998*, 1998.

[3] M. Capps, D. McGregor, D. Brutzman, and M. Zyda, "NPSNET-V. A new beginning for dynamically extensible virtual environments," *IEEE Computer Graphics and Applications*, vol. 20, no. 5, pp. 12–15, 2000.

[4] D. Kushner, "The wizardry of Id," *IEEE Spectrum*, vol. 39, no. 8, pp. 42–47, 2002.

[5] J. Smed, T. Kaukoranta, and H. Hakonen, "A review on networking and multiplayer computer games," Tech. Rep. 454, Turku Centre for Computer Science, University of Turku, Turku, Finland, 2002.

[6] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of networking in multiplayer computer games," *The Electronic Library*, vol. 20, no. 2, pp. 87–97, 2002.

[7] D. Delaney, T. Ward, and S. McLoone, "On consistency and network latency in distributed Interactive applications: a survey—part I," *Presence: Teleoperators and Virtual Environments*, vol. 15, no. 2, pp. 218–234, 2006.

[8] X. Qin, "Delayed consistency model for distributed interactive systems with real-time continuous media," *Journal of Software*, vol. 13, no. 6, pp. 1029–1039, 2002.

[9] S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison-Wesley, New York, NY, USA, 1st edition, 1999.

[10] Y. Yu, Z. Li, L. Shi, Y.-C. Chen, and H. Xu, "Network-aware state update for large scale mobile games," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 563–568, Honolulu, Hawaii, USA, August 2007.

[11] W. R. Johnson, T. W. Mastaglio, and P. D. Peterson, "The close combat tactical trainer program," in *Proceedings of the 25th Winter Simulation Conference (WSC '93)*, pp. 1021–1029, Los Angeles, Calif, USA, December 1993.

[12] J. D. Delaney, *Latency reduction in distributed interactive applications using hybrid strategy-based models*, Ph.D. thesis, Department of Electronic Engineering, National University of Ireland, Maynooth, Maynooth, Ireland, 2004.

[13] A. McCoy, T. Ward, S. McLoone, and D. Delaney, "Multistep-ahead neural-network predictors for network traffic reduction in distributed interactive applications," *ACM Transactions on Modeling and Computer Simulation*, vol. 17, no. 4, article 16, pp. 1–30, 2007.

[14] A. McCoy, T. Ward, S. McLoone, and D. Delaney, "Using neural-networks to reduce entity state updates in distributed interactive applications," in *Proceedings of the 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing (MLSP '06)*, pp. 295–300, Maynooth, Ireland, September 2007.

[15] C. Diot and L. Gautier, "Distributed architecture for multiplayer interactive applications on the Internet," *IEEE Network*, vol. 13, no. 4, pp. 6–15, 1999.

[16] J. J. LaViola, "A testbed for studying and choosing predictive tracking algorithms in virtual environments," in *Proceedings of the Workshop on Virtual Environments*, pp. 189–198, Zurich, Switzerland, May 2003.

[17] J. C. S. Lui, "Constructing communication subgraphs and deriving an optimal synchronization interval for distributed virtual environment systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 5, pp. 778–792, 2001.

[18] S. Zhou, W. Cai, B.-S. Lee, and S. J. Turner, "Time-space consistency in large-scale distributed virtual environments," *ACM Transactions on Modeling and Computer Simulation*, vol. 14, no. 1, pp. 31–47, 2004.

[19] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, "The mutual information: detecting and evaluating dependencies between variables," *Bioinformatics*, vol. 18, supplement 2, pp. S231–S240, 2002.

[20] J. Jeong, J. C. Gore, and B. S. Peterson, "Mutual information analysis of the EEG in patients with Alzheimer's disease," *Clinical Neurophysiology*, vol. 112, no. 5, pp. 827–835, 2001.

[21] E. Frias-Martinez, G. Magoulas, S. Chen, and R. Macredie, "Modeling human behavior in user-adaptive systems: recent advances using soft computing techniques," *Expert Systems with Applications*, vol. 29, no. 2, pp. 320–329, 2005.

[22] W. Li, "Mutual information functions versus correlation functions," *Journal of Statistical Physics*, vol. 60, no. 5-6, pp. 823–837, 1990.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2006.

[24] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003.

[25] M. S. Roulston, "Estimating the errors on measured entropy and mutual information," *Physica D*, vol. 125, no. 3-4, pp. 285–294, 1999.

[26] A. B. McCoy, *Data-driven modelling approaches for network traffic reduction in distributed interactive applications*, Ph.D. thesis, Department of Electronic Engineering, National University of Ireland, Maynooth, Maynooth, Ireland, 2007.

[27] B.-S. Lee, W. Cai, S. J. Turner, and L. Chen, "Adaptive dead reckoning algorithms for distributed interactive simulation," *International Journal of Simulation Systems, Science & Technology*, vol. 1, pp. 21–34, 2000.

[28] D. Delaney and T. Ward, "A Java tool for exploring state estimation using the Kalman filter," in *Proceedings of IEEE Irish Signals and Systems Conference (ISSC '04)*, pp. 679–684, Belfast, Northern Ireland, June-July 2004.

[29] T. P. Duncan and D. Gračanin, "Pre-reckoning algorithm for distributed virtual environments," in *Proceedings of the 35th Winter Simulation Conference (WSC '03)*, vol. 2, pp. 1086–1093, New Orleans, La, USA, December 2003.

[30] M. A. Bassiouni, M.-H. Chiu, M. Loper, M. Garnsey, and J. Williams, "Performance and reliability analysis of relevance filtering for scalable distributed interactive simulation," *ACM Transactions on Modeling and Computer Simulation*, vol. 7, no. 3, pp. 293–331, 1997.

[31] K. Chen and T. V. Ramabadran, "Near-lossless compression of medical images through entropy-coded DPCM," *IEEE Transactions on Medical Imaging*, vol. 13, no. 3, pp. 538–548, 1994.

[32] International Standard, ISO/IEC 11172, "Information technology—Generic Coding of Moving Pictures and Associated Audio: Systems," 1994.