

Formalizing a Framework for Dynamic Hybrid Strategy Models in Distributed Interactive Applications

Aaron McCoy, Tomás Ward, Seamus McLoone and Declan Delaney

*Department of Electronic Engineering,
National University of Ireland Maynooth,
Maynooth, Co. Kildare,
Republic of IRELAND
E-mail: amccoy@eeng.nuim.ie*

Predictive contract mechanisms such as dead reckoning are widely employed to support scalable remote entity modelling in Distributed Interactive Applications (DIAs). By employing a form of controlled inconsistency, a reduction in network traffic is achieved. Previously, we have proposed the Dynamic Hybrid Strategy Model (DHSM) as an extension to the concept of dead reckoning that adaptively selects extrapolation models based on the use of local performance criteria. In this paper, we formalize the notion of the DHSM as a generalized framework for network traffic reduction in DIAs, alongside a set of consistency metrics for use as local performance criteria.

Keywords – distributed interactive applications, predictive contract mechanisms, dead reckoning, consistency, scalability.

I INTRODUCTION

A Distributed Interactive Application (DIA) is a distributed virtual reality system through which individuals can share information via individual and collaborative interaction with each other and their environment [1]. Distributed Interactive Applications offer the realization of simulated virtual worlds that embody a modern extension of communication, encompassing the concepts of *shared time*, *shared space* and *shared presence* [2]. The definition of a DIA encompasses a diverse range of applications that have seen rapid advances in technology and global popularity due to the widespread availability and ease-of-use of the Internet [3].

The two primary factors limiting the large-scale deployment of a DIA are network latency and network bandwidth. Network latency refers to the delay in communication between two end-points, while network bandwidth refers to the rate at which data can be communicated per unit time between two end-points. High network latency and low network bandwidth capacity represent the largest contributors to the difficulties faced by DIAs in maintaining and supporting: (a) shared state consistency, (b) potential scalability, and (c) real-time interactivity.

In this paper, we are concerned with the use of predictive contract mechanisms for the reduction of network traffic in DIAs, including the well-known dead reckoning algorithms formally defined in the

IEEE Standard for Distributed Interactive Simulation (DIS) [4]. Traditional dead reckoning mechanisms often ignore available contextual information that may be influential to the state of an entity, sacrificing remote predictive accuracy in favour of low resource and computational overhead [2]. Previously, we have proposed an extension of dead reckoning, known as the Dynamic Hybrid Strategy Model (DHSM), that builds upon the foundation of the Hybrid Strategy Model (HSM) technique [5, 6]. Both the HSM and the DHSM are hybrid predictive contract techniques that dynamically select remote extrapolation models based on local evaluation of current entity dynamics.

In this paper, we formalize the concept of the DHSM as a generalized framework for the reduction of network traffic in DIAs, providing a specification and associated algorithm for implementation in real-world scenarios. We formalize a set of corresponding consistency metrics that can be used as performance criteria for the dynamic (and adaptive) selection of extrapolation models, including the use of time-space consistency metrics as a combination of spatial and temporal threshold dimensions.

The rest of this paper is organized as follows. In Section 2, we review work relating to both predictive contract mechanisms and measures of consistency. In Section 3, we formally define consistency metrics in relation to dead reckoning mechanisms. In Section 4, we present our proposed DHSM framework. Section 5 offers conclusions and suggestions for future work.

II RELATED WORK

a) Predictive Contract Mechanisms

Dead reckoning is the most widely employed form of predictive contract mechanism in current use. It was introduced during the development of SIMNET [7] and subsequently formally defined within the IEEE Standard for Distributed Interactive Simulation (DIS) [4]. Controlling hosts interacting within a DIA are required to maintain two parallel models for each entity under their control – a high-fidelity dynamical model that represents true entity state as determined through autonomous local control, and a low-fidelity dynamics model representing remotely approximated entity state. By periodically comparing the deviation between the two models against an error threshold or violating a transmission timeout criterion, controlling hosts can locally determine when state information is required to be replicated to remote hosts. Hence, by employing a form of *controlled inconsistency*, dead reckoning achieves a reduction in network traffic.

The methodology behind dead reckoning (that is to say, the distribution of low-fidelity models to remote hosts and the subsequent (non-periodic) update of those models based on local computation performed by the controlling host) is quite often referred to as the ‘*players and ghosts*’ paradigm [8]. In this case, it is assumed that each entity is owned by a single controlling host (the *player*), who is required to update the remote simulations of that entity (the *ghosts*) when the deviation between local and remote state exceeds the given threshold(s).

Dead reckoning mechanisms consist of two main components, namely *prediction* and *convergence* [2]. Prediction algorithms (often known as *extrapolation models*) are defined in Section IV. They specify how entity state is remotely extrapolated between entity state updates (ESUs). Convergence algorithms define how remote entity state is smoothly corrected due to errors arising from inaccurate extrapolation. Detailed discussion of convergence algorithms is beyond the scope of this paper.

As previously described, dead reckoning schemes ignore available contextual information that may be influential to the state of an entity. To compensate, a number of adaptive extrapolation schemes have been proposed throughout the literature, aimed at reducing the error associated with remote extrapolation. These schemes can be distinguished according to the nature of computation associated with the adaptive process:

1. *Local Computation Schemes*: Local computation schemes require controlling hosts to select an appropriate extrapolation model for each entity under their control, distributing application-wide computational overhead *linearly* with the number of hosts [5, 6, 9, 10, 11];
2. *Remote Computation Schemes*: Remote hosts are required to adaptively select extrapolation models

for each remotely modelled entity. Computational overhead is distributed throughout the application *quadratically* with the number of hosts [12, 13].

In order to determine when to send a new entity state update (ESU), dead reckoning mechanisms require an explicit means of measuring the current degree of dynamic shared state consistency between the high-fidelity and low-fidelity entity models (in a practical sense, the level of inconsistency accrued between the local and remote representations of an entity will be higher than that estimated locally due to the presence of network latency).

Traditionally, dead reckoning mechanisms have relied on spatial-based and temporal-based metrics as orthogonal quantities, merely issuing an entity state update (ESU) when either has been exceeded (these metrics are used under the guise of ‘*error threshold*’ and ‘*transmission timeout*’, respectively). The error threshold is designed to bound the remote predictive error associated with the extrapolation of entity state, while the transmission timeout seeks to guarantee a minimum frequency of ESU transmission at the cost of potentially redundant bandwidth usage. Used in isolation, temporal-metrics are quite often referred to as *frequent state regeneration* with *heartbeat packets* [2, 14]. In order to optimise the selection of these parameters, various adaptive error threshold schemes have been proposed throughout the literature [8, 10, 15, 16, 17, 18, 19], along with a series of techniques for reducing the reliance on the transmission timeout criterion [14].

b) Measures of Consistency

In the literature, *drift distance* has been defined as a metric for evaluating the consistency in a DIA that represents absolute spatial difference between local and remote entity models [20]. *Phase difference* has been defined as a consistency metric that represents the temporal difference between the rendering times for local and remote entity models [21]. *Export error* has been defined as the deviation in physical time of the remote entity model from the local entity model [22]. *Deviation* and *uncertainty* have been defined as absolute spatial difference and the size of the area in which an entity could reside in, respectively [17].

More recently, the use of a formalized time-space consistency metric has been proposed to evaluate the effect of inconsistency on user perception in a DIA, and it is given by the following equation [23]:

$$\Omega = \begin{cases} 0, & \text{if } |\Delta(t)| < \varepsilon, \\ \int_{t_0}^{t_0+\tau} |\Delta(t)| dt, & \text{if } |\Delta(t)| \geq \varepsilon, \end{cases} \quad (1)$$

where $\Delta(t)$ is the spatial difference between local and remote entity models, ε is the minimum perceivable error (from a user perspective), t_0 is the time at which

the difference starts, and τ is the duration for which the difference persists. When Ω equals zero, *absolute consistency* occurs. Time-space consistency metrics have been proposed for use as local error thresholds within dead reckoning mechanisms [24].

In the next section, we formalize the concepts of spatial, temporal, and time-space consistency metrics within the context of dead reckoning mechanisms. It should be noted that our proposed set of definitions are specific to our particular application domain, and are not intended as general metrics. This means, for instance, that our definition for temporal consistency is independent of other definitions proposed in such papers as [25], which typically deal with intervals of validity for sampled variables.

III CONSISTENCY METRICS

a) Concepts of Time

We view a DIA as a discrete domain where time is modelled as a monotonically increasing sequence of time instances bounded by a minimal element $t_0 = 0$:

$$T = \{t_i\}, \forall i \in \{0\} \cup \mathfrak{S}^+ \quad (2)$$

Given the above model of time, we define the closed interval between any pair of time instances $[t_m, t_n]$ as:

$$T_{[t_m, t_n]} = \{t_i \mid t_i \in T, t_m \leq t_i \leq t_n\} \quad (3)$$

We define a temporal-metric τ for referencing wall-clock time as the *constant duration* (as measured in seconds) *between successive time instances*:

$$\tau = t_i - t_{i-1}, \forall i \in \mathfrak{S}^+ \wedge t_i \in T \quad (4)$$

Hence, a given time instance t_i can be de-referenced to the corresponding wall-clock (physical) time by:

$$t_i = i\tau, \forall t_i \in T \quad (5)$$

Equations (2-5) provide a time-base for the definition of dead reckoning consistency metrics, as proposed throughout the following series of subsections.

b) Spatial Consistency

We define a measure of *spatial consistency* for entity \mathbf{E} interacting in a DIA at time t_i as the instantaneous, absolute positional difference between the local low-fidelity and high-fidelity models (denoted as \mathbf{x}^{low} and \mathbf{x}^{high} respectively) for that entity:

$$\Delta_{t_i}^{\mathbf{E}} = \left| \mathbf{x}_{t_i}^{high} - \mathbf{x}_{t_i}^{low} \right| \quad (6)$$

Hence, for the purposes of dead reckoning, we say an entity \mathbf{E} is *spatially consistent* (SC) at time t_i if and

only if the measured spatial consistency for the entity at that time is less than or equal to a bounded spatial error threshold δ_{Δ} :

$$SC_{t_i}^{\mathbf{E}} \Leftrightarrow \Delta_{t_i}^{\mathbf{E}} \leq \delta_{\Delta} \quad (7)$$

c) Temporal Consistency

We define a measure of *temporal consistency* for an entity \mathbf{E} interacting within a DIA as the total elapsed wall-clock duration (measured in seconds) during the closed interval $[t_r, t_i]$, where t_r is an arbitrarily chosen past time instance such that $r < i$:

$$\Phi_{[t_r, t_i]}^{\mathbf{E}} = t_i - t_r = (i - r)\tau \quad (8)$$

Hence, for the purposes of dead reckoning, we say an entity \mathbf{E} is *temporally consistent* (TC) at time t_i if and only if the measured temporal consistency for the entity during the closed interval $[t_{\lambda}^{\mathbf{E}}, t_i]$ is less than or equal to a bounded temporal error threshold δ_{Φ} , such that $t_{\lambda}^{\mathbf{E}}$ represents the transmission time of the most recent entity state update (ESU) for that entity:

$$TC_{t_i}^{\mathbf{E}} \Leftrightarrow \Phi_{[t_{\lambda}^{\mathbf{E}}, t_i]}^{\mathbf{E}} \leq \delta_{\Phi} \quad (9)$$

d) Time-Space Consistency

We define a measure of *time-space consistency* for an entity \mathbf{E} interacting in a DIA as the area under the graph of spatial consistency over time throughout the closed interval $[t_r, t_i]$, where t_r is an arbitrarily chosen past time instance such that $r < i$:

$$\Omega_{[t_r, t_i]}^{\mathbf{E}} = \frac{\tau}{2} \sum_{j=r+1}^i (\Delta_{t_j}^{\mathbf{E}} + \Delta_{t_{j-1}}^{\mathbf{E}}) \quad (10)$$

Hence, for the purposes of dead reckoning, we say an entity \mathbf{E} is *temporally-spatially consistent* (TSC) at time t_i if and only if the measured time-space consistency for the entity during the closed interval $[t_{\lambda}^{\mathbf{E}}, t_i]$ is less than or equal to a bounded temporal-spatial error threshold δ_{Ω} , such that $t_{\lambda}^{\mathbf{E}}$ represents the time of transmission of the most recent entity state update (ESU) for that entity:

$$TSC_{t_i}^{\mathbf{E}} \Leftrightarrow \Omega_{[t_{\lambda}^{\mathbf{E}}, t_i]}^{\mathbf{E}} \leq \delta_{\Omega} \quad (11)$$

IV DHSM FRAMEWORK

a) Specification

A Dynamic Hybrid Strategy Model (DHSM) D can be specified as an ordered quadruple of the following form:

$$D = \langle M, m_0, \rho, H \rangle \quad (12)$$

wherein:

- M is a finite, non-empty set of n candidate extrapolation models, only one of which is ever *active* at any one time:

$$M = \{m_1, m_2, \dots, m_n\} \quad (13)$$

- $m_0 \in M$ is the *default active* extrapolation model;
- ρ is a performance function used to rate the accuracy of state estimation across the set of extrapolation models M , such that:

$$\rho(M) = \{\rho(m_1), \rho(m_2), \dots, \rho(m_n)\} \quad (14)$$

- H is a model transition table (function) that maps from $M \times \rho \rightarrow M$, and can be represented as a set of ordered triples of the following form:

$$H = \langle m_i, \rho(M), m_j \rangle \quad (15)$$

wherein:

- $m_i \in M$ is the *current active* extrapolation model at time t_n ;
- $m_j \in M$ is the *next active* (selected) extrapolation model at time t_{n+1} .

b) Extrapolation Models

Given the discrete time-base defined previously by Equations (2-5), we can specify the general form for an extrapolation model $m_i \in M$ as a direct functional mapping from both the current and k previous states, actions, and environmental (external) influences on the future (estimated) state q steps ahead for an entity \mathbf{E} interacting in a DIA at time t_i :

$$\mathbf{s}_{t_{i+q}}^{\mathbf{E}} = f_{m_i}(\mathbf{s}_{[t_{i-k}, t_i]}^{\mathbf{E}}, \mathbf{a}_{[t_{i-k}, t_i]}^{\mathbf{E}}, \mathbf{e}_{[t_{i-k}, t_i]}^{\mathbf{E}}) \quad (16)$$

wherein:

- $\mathbf{s}_{t_{i+q}}^{\mathbf{E}}$ is the estimated entity state q steps ahead;
- $\mathbf{s}_{[t_{i-k}, t_i]}^{\mathbf{E}}$ are the current and k past entity states;
- $\mathbf{a}_{[t_{i-k}, t_i]}^{\mathbf{E}}$ are the current and k past entity actions;
- $\mathbf{e}_{[t_{i-k}, t_i]}^{\mathbf{E}}$ are the current and k past influences.

The functional form of f defines how entity state is extrapolated remotely, while its inputs define what information must be communicated over the network in the form of entity state updates (ESUs) (due to the fact that remote hosts must have the information to perform the extrapolation). Dead reckoning schemes

typically extrapolate over single steps ($q = 1$), ignore action and environmental variables, and traditionally define state vectors in terms of multi-dimensional location, velocity, and acceleration components. Equation (16) implicitly defines both one-step ($k = 0$) and multi-step ($k > 0$) extrapolation models [10].

A detailed discussion on how to define the form of the extrapolation function f is beyond the scope of this paper. The problem may be formulated as one of inference in a temporal probability model, and in this respect has been well studied in the field of artificial intelligence using statistical models such as Dynamic Bayesian Networks (DBNs) and Kalman Filters [26].

c) Performance Function

The role of the performance function ρ is to assign a *rating* (or *score*) for each extrapolation model based on overall predictive performance (i.e. the accuracy of entity state estimation) between consecutive entity state update (ESU) transmissions. Thus, for an entity \mathbf{E} interacting in a DIA at time t_i , ρ is defined over the interval $[t_\lambda^{\mathbf{E}}, t_i]$, where $t_\lambda^{\mathbf{E}}$ is the time of transmission of the most recent ESU for that entity. Naturally, the obvious choice would be to simply utilize the current consistency metric to rate model performance. In this respect, a temporal consistency metric is independent of entity dynamics, and has no performance function analogue. For spatial consistency metrics, scores can be assigned using the average spatial consistency for an entity \mathbf{E} over the interval $[t_\lambda^{\mathbf{E}}, t_i]$, given by:

$$\rho(M) = \bar{\Delta}_{[t_\lambda^{\mathbf{E}}, t_i]}^{\mathbf{E}} = \frac{1}{(i - \lambda)} \sum_{j=\lambda}^i \Delta_{t_j}^{\mathbf{E}} \quad (17)$$

For time-space consistency metrics, scores can be assigned using Equation (10). Performance functions and consistency metrics are not mutually exclusive – hence the distinction we have made between the two. For instance, it is entirely reasonable to utilize spatial consistency metrics for ESU generation, alongside a time-space performance function for assigning scores to models.

d) Model Transition Table

The model transition table H utilizes the performance function ρ to dynamically select *active* extrapolation models, as shown in Figure 1. It accepts the *current active* extrapolation model (denoted as $m_i \in M$) and the performance function evaluated over the set of all extrapolation models (denoted as $\rho(M)$), and outputs the *next active* extrapolation model (denoted as $m_j \in M$). Technically, it can be called at any time t_i and for any arbitrary performance function interval $[t_m, t_n]$ to select a new *active* extrapolation model. In general, however, we only call it whenever a new ESU needs to be transmitted from the local host to remote hosts for an entity \mathbf{E} (i.e. an interval $[t_\lambda^{\mathbf{E}}, t_i]$, where t_i is the present time and $t_\lambda^{\mathbf{E}}$ the time of most recent ESU).

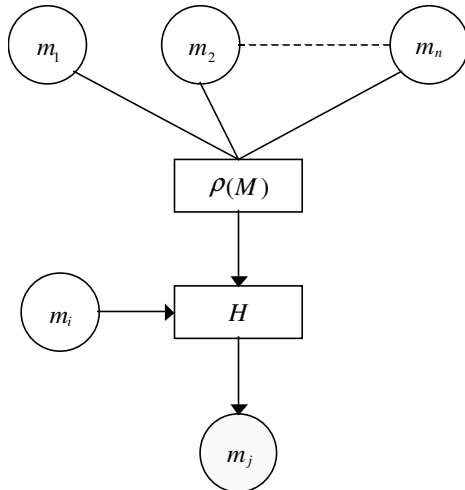


Figure 1: Selecting a new *active* extrapolation model.

The model transition table H can be viewed as a black-box, whose role is to define the set of rules that govern the dynamic selection of *active* extrapolation models. In a basic form, it simply chooses the model that exhibits the best score, as determined using the performance function. Issues may arise, however, in situations where multiple models exhibit an identical performance rating. In such cases, the responsibility for resolving the contention and specifying the model precedence also falls to the model transition table H , and this must be dealt with by defining any ancillary model selection conditions, parameters, functions or criteria that must be met.

e) Algorithmic Description

The operation of the DHSM, from the perspective of the *local* (controlling) *host* for an entity \mathbf{E} interacting in a DIA, can be described as follows:

1. M is distributed to *remote hosts*, alongside a state update to initialise \mathbf{E} and inform them of m_0 . m_0 is set as the *default active* extrapolation model;
2. At each time instance t_i during the simulation, the local host:
 - a) Computes the current high-fidelity entity state based on user-input;
 - b) Estimates the current low-fidelity entity state for each model $m_i \in M$ in *parallel*;
 - c) Quantifies the prediction error for each model $m_i \in M$ using the given consistency metric(s);
 - d) Checks for error threshold violation (i.e. ESU transmission) using the *active model only*;
 - e) Promptly proceeds to Step 3 (see below) if an error threshold violation is detected;
3. If an error threshold violation is detected, an ESU is required to be transmitted to remote hosts, and the local host:
 - a) Assigns a score to each model $m_i \in M$ using the performance function, as described above;

- b) Uses the model transition table to select a new *active* model, as described previously;
- c) Sends the identifier and required state data for the new *active* model to remote hosts ASAP;
- d) Resets the score and state estimators for each model $m_i \in M$, and proceeds back to Step 2.

For simplicity of description, the algorithm given above assumes a single entity \mathbf{E} . Handling multiple entities is a trivial task, however – the only real issue is whether to define the components of the DHSM in an *entity-centric* or *host-centric* domain. In an *entity-centric domain*, components are defined on a one-to-one basis for each individual entity, offering greater potential flexibility at the cost of enlarged processing and memory overhead. In a *host-centric domain*, the components are shared among sub-ordinates who are controlled by the host – so for instance, performance functions and sets of extrapolation models are shared among entities. The choice is application-dependent.

V CONCLUSIONS

In this paper, we have defined a formal specification for the Dynamic Hybrid Strategy Model (DHSM) as a general framework for the reduction of network traffic in DIAs. The DHSM expands on the concepts of traditional dead reckoning and incorporates theory relating to the field of Switching Kalman Filters [26], namely the use of multiple state estimators in parallel with one another. By scoring model performance as a measure of consistency and subsequently rating them relative to one another, the DHSM attempts to make an informed decision regarding the suitable selection of *remote* extrapolation models based on their recent accuracy in estimating future entity state *locally*. The dynamic model switching feature enables the DHSM to adapt to perceived changes in entity behaviour, all computed on a local-basis by each controlling host.

Ultimately, the utility of the DHSM for providing a further reduction in network traffic over traditional dead reckoning mechanisms depends on the capacity of the set of extrapolation models that comprise it to accurately estimate entity state. The DHSM assumes that recent entity behaviour provides an indication of future entity behaviour on a comparatively long-term basis. Given this limitation, future work will involve the investigation of alternative extrapolation models based on the techniques of user modelling [27].

VI ACKNOWLEDGEMENT

This material is based upon works supported by the Science Foundation of Ireland and Enterprise Ireland under grant no. IRCSET/SC/04/CS0289, and also by Enterprise Ireland under grant no. SC/2002/129.

VII REFERENCES

- [1] E. F. Churchill, D. N. Snowdon and A. J. Munro, *Collaborative Virtual Environments: Digital Places*

- and Spaces for Interaction. London, United Kingdom: Springer-Verlag, 2001.
- [2] S. K. Singhal and M. J. Zyda, *Networked Virtual Environments: Design and Implementation*. New York, New York: Addison-Wesley, ACM Press SIGGRAPH Series, 1999.
 - [3] A. McCoy, D. Delaney and T. Ward, "Game-State Fidelity Across Distributed Interactive Games," *ACM Crossroads*, vol. 9.4 (Networking Issue), pp. 4-9, Summer 2003.
 - [4] *IEEE Standard for Distributed Interactive Simulation - Application Protocols*, IEEE Standard 1278.1-1995, 1995.
 - [5] D. Delaney, T. Ward and S. McLoone, "Reducing Update Packets in Distributed Interactive Applications using a Hybrid Model," in *Proc. ISCA 16th International Conference on Parallel and Distributed Computing Systems*, Reno, Nevada, USA, 2003, pp. 417-422.
 - [6] A. McCoy, D. Delaney, S. McLoone and T. Ward, "Dynamic Hybrid Strategy Models for Networked Multiplayer Games," in *Proc. 19th European Conference on Modelling and Simulation (ECMS 2005)*, Riga, Latvia, June 2005, pp. 727-732.
 - [7] D. C. Miller and J. A. Thorpe, "SIMNET: The Advent of Simulator Networking," *Proceedings of the IEEE*, vol. 83, no. 8, pp. 1114-1123, August 1995.
 - [8] B. Blau, C. E. Hughes, J. M. Moshell and C. Lisle, "Networked Virtual Environments," in *Proc. 1992 ACM International Symposium on Interactive 3D Graphics*, Cambridge, Massachusetts, USA, 1992, pp. 157-160.
 - [9] T. P. Duncan and D. Gracanin, "Pre-Reckoning Algorithm for Distributed Virtual Environments," in *Proc. 2003 Winter Simulation Conference*, Fairmount Hotel, New Orleans, Louisiana, USA, 2003, pp. 1086-1093.
 - [10] B.-S. Lee, W. Cai, S. J. Turner and L. Chen, "Adaptive Dead Reckoning Algorithms for Distributed Interactive Simulation," *International Journal of Simulation*, vol. 1, no. 1-2, pp. 21-34, December 2000.
 - [11] J. Marsh, S. Pettifer and A. West, "A Technique for Maintaining Continuity of Experience in Networked Virtual Environments", in *Proc. UKVRSIG '99*, The University of Salford, Greater Manchester, United Kingdom, September 1999.
 - [12] Y.-P. Chui and P.-A. Heng, "Attitude Dead Reckoning in a Collaborative Virtual Environment using Cumulative Polynomial Extrapolation of Quaternions," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 15 (Special issue on Distributed Simulation and Real-Time Applications), pp. 1575-1599, December 2004.
 - [13] S. K. Singhal and D. R. Cheriton, "Exploring Position History for Efficient Remote Rendering in Networked Virtual Reality," *Presence: Teleoperators and Virtual Environments*, vol. 4, no. 2, pp. 169-193, Spring 1995.
 - [14] D. J. Van Hook, J. O. Calvin and J. E. Smith, "Data Consistency Mechanisms to Support Distributed Simulation," in *Proc. 12th Workshop on Standards for the Interoperability of Distributed Simulations*, Institute for Simulation and Training, Orlando, Florida, USA, 1995, pp. 797-806.
 - [15] J. O. Calvin, J. Seeger, G. D. Troxel and D. J. V. Hook, "STOW Realtime Information Transfer and Networking System Architecture," in *Proc. 12th Workshop on Standards for the Interoperability of Distributed Simulations*, Institute for Simulation and Training, Orlando, Florida, USA, March 1995, pp. 343-353.
 - [16] K.-H. Shim and J.-S. Kim, "A Dead Reckoning Algorithm with Variable Threshold Scheme in Networked Virtual Environment," in *Proc. 2001 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2001)*, Tucson, Arizona, USA, 2001, pp. 1113-1118 (vol. 2).
 - [17] O. Wolfson, L. Jiang, A. P. Sistla, S. Chamberlain, N. Rish and M. Deng, "Databases for Tracking Mobile Units in Real Time," in *Proc. 7th International Conference on Database Theory (ICDT '99)*, Jerusalem, Israel, 1999, pp. 169-186.
 - [18] S.-J. Yu and Y.-C. Choy, "An Adaptive Dead Reckoning Algorithm using Update Lifetime," *Virtual Reality*, vol. 5, no. 3, pp. 132-148, September 2000 2000.
 - [19] M.-J. Zhang and N. D. Georganas, "An Orientation Update Message Filtering Algorithm in Collaborative Virtual Environments," *Journal of Computer Science and Technology (JCST)*, vol. 19, no. 3, pp. 423, May 2004 2004.
 - [20] C. Diot and L. Gautier, "A Distributed Architecture for Multiplayer Applications on the Internet," *IEEE Network*, vol. 13, no. 4, pp. 6-15, July/August 1999.
 - [21] 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, September / October 2001 2001.
 - [22] S. Aggarwal, H. Banavar, A. Khandelwal, S. Mukherjee and S. Rangarajan, "Accuracy in Dead-Reckoning Based Distributed Multi-Player Games," in *Proc. 3rd ACM SIGCOMM Workshop on Network and System Support for Games (NetGames '04)*, Portland, Oregon, USA, 2004, pp. 161-165.
 - [23] 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, January 2004 2004.
 - [24] D. J. Roberts, D. Marshall, S. McLoone, D. Delaney, T. Ward and R. Aspin, "Exploring the Use of Local Consistency Measures as Thresholds for Dead Reckoning Update Packet Generation," in *Proc. 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2005)*, Montreal, Canada, 2005, pp. 195-202.
 - [25] M. Xiong, K. Ramamritham, J. A. Stankovic, D. Towsley and R. Sivasankaran, "Scheduling Transactions with Temporal Constraints: Exploiting Data Semantics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1155-1166, September/October 2002.
 - [26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Prentice Hall Series in Artificial Intelligence (Pearson Education, Inc.), 2003.
 - [27] I. Zukerman and D. W. Albrecht, "Predictive Statistical Models for User Modeling," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 5-18, March 2001.