DEALING WITH THE EFFECT OF PATH CURVATURE ON CONSISTENCY OF DEAD RECKONED PATHS IN NETWORKED VIRTUAL ENVIRONMENTS

Damien Marshall*°, Dave Roberts*, Declan Delaney°, Seamus McLoone°, Tomas Ward°

* Centre for Virtual Environments
Business House,
University of Salford,
Salford, M5 4WT
United Kingdom

damienm@cs.nuim.ie,
d.j.roberts@salford.ac.uk

°National University of Ireland Maynooth
Maynooth,
Co. Kildare,
Ireland

decland@cs.nuim.ie
seamus.mcloone@eeng.nuim.ie
tomas.ward@eeng.nuim.ie

Abstract

Several techniques exist which reduce network bandwidth consumption and thus limit the effects of network latency in networked virtual environments. Dead reckoning is one such technique that has been widely adopted for use with dynamic entities. However, in previous work we have shown that the use of a spatial threshold alone in dead reckoning can result in unbounded absolute inconsistency. A novel hybrid threshold was proposed that combined a spatial threshold together with an absolute consistency metric to impose an upper bound on absolute inconsistency under all circumstances. This was verified through a simulation of typical movement in a computer racing game. This paper extends this work by investigating the problem in more detail and verifying the hybrid threshold solution in live trials across the Internet. In particular the relationship between curvature of movement and absolute consistency is explored. The experimental trials allow a comparative analysis of how users behave when different threshold types are used under varying degrees of curvature. This work provides further justification for the use of a hybrid threshold approach when dead reckoning is employed in networked virtual environments.

1. Introduction

Collaboration and competition are important factors of networked virtual environments. Despite the obvious conflict between them, they both require a certain level of consistency in order for the interaction to be fruitful and compelling. Consistency has been defined in many ways [1]. For the purposes of this paper, consistency refers to three different aspects – spatial consistency, temporal consistency and absolute consistency. A movement trajectory may be spatially consistent with the original movement without being temporally consistent. This is often the case in networked virtual environments that do not consider clock synchronisation or network delay. Absolute consistency is a more meaningful measure for real time applications as it describes synchronisation in terms of both space and time [2].

One of the key limiting factors in achieving consistency is network latency. Network latency introduces delays that arise for a number of reasons, including packet routing delays, packet processing delays, network congestion and the inherent limit on the speed of transmission of information imposed by the physical speed of the medium.

A popular method for overcoming network delays and reducing network traffic for dynamic entities is the IEEE DIS standard dead reckoning mechanism [3]. The mechanism operates as follows. Each participant transmits position and velocity data in an update packet whenever a model of their movement and their actual movement differ by a threshold value. All other participants then use this data to model the behaviour of each entity. In the DIS standard this error threshold is based on the spatial distance between the two positions.

In previous work, it was demonstrated that the conventional approach of using a purely spatial threshold for dead reckoning can lead to bounded spatial inconsistency but unbounded absolute inconsistency [4]. A novel threshold metric to replace the existing dead reckoning spatial error threshold was proposed. This threshold uses local absolute consistency measures in determining when an update is required, and was shown to result in bounded local absolute inconsistency but unbounded spatial inconsistency. A hybrid threshold combining both a spatial threshold and the absolute consistency metric results in both bounded spatial inconsistency and bounded absolute inconsistency.

In this paper we further explore the use of various local inconsistency measures as thresholds for the dead reckoning algorithm. In particular the issue of unbounded spatial consistency and path curvature is explored when using a threshold based on the absolute consistency metric.

The key contribution here is the mathematical exploration of the relationship between path curvature and absolute inconsistency. This deepens our understanding of the inconsistency metric threshold in addition to providing additional motivation for the hybrid threshold approach. In addition a series of experiments are performed using an industry standard games engine called Torque to test the hybrid and spatial thresholds under varying conditions of curvature.

The rest of the paper is laid out as follows. Section 2 provides a background to the research area, and the novel dead reckoning error threshold is introduced. In Section 3, it is shown how varying path curvature can lead to unbounded absolute inconsistency when a spatial error threshold is employed with dead reckoning. Section 4 describes the experiments performed conducted using various threshold metrics. The results of these experiments are discussed in section 5 and the paper concludes in Section 6 with some final comments.

2. Background

Attempts at measuring consistency have been attempted by several authors. Diot et. al use a metric known as "drift distance" to measure the inconsistency of their MiMaze application [5]. Drift distance is the spatial difference between the actual and remote entity position at each time step. This measurement ignores the duration of the inconsistency (and in any case is impossible to obtain in real time). On the other hand, Lui proposes a measure which is based on time only [6]. The author proposes inconsistency to be the difference between the start times of rendering of the same changes to the world at different nodes. This is known as "phase difference". Another interesting factor of consistency that has been examined is that of semantic inconsistency. According to Correa et al., semantic consistency pertains to the meaning of the data present in the virtual environment [7]. The authors measure semantic consistency in order to reduce network traffic that needs to be communicated amongst nodes with heterogeneous resources.

Zhou et al. provide a method of measuring inconsistency in both the time and space domain [8]. They use this metric to measure the absolute consistency arising from a networked virtual environment simulation before execution time. The inconsistency metric is given in Equation 1.

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

where

$\Delta(t)$ is the difference between the position of a local object and its remote replication at time t;

t_o is the start time of the inconsistency and τ is the duration;

ε is the minimum perceivable distance.

In previous work it was demonstrated, using Zhou's metric, that absolute inconsistency is unbounded when a spatial threshold is employed with dead reckoning [4]. To rectify this, a new dead reckoning threshold metric was proposed [4]. This metric employs Zhou's measure of absolute inconsistency in determining when a dead reckoning update should be transmitted. It was shown that the use of this metric places a bound on absolute inconsistency but could result in unbounded spatial inconsistency.

The following section explores, in greater detail, why this unbounded spatial inconsistency occurs by deriving a relationship between the inconsistency metric and path curvature.

3. Path curvature and consistency

l – distance travelled in time interval
v – velocity
d – distance between actual and model positions
k – path curvature
P_A – actual position
P_M – model position

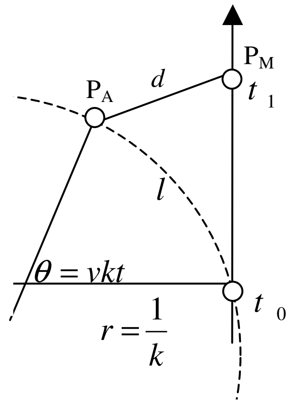


Figure 1. Actual and dead reckoning model position

Consider the case where the curvature, *k*, and velocity, *v*, of the actual path taken by an entity are constant – see Figure 1. The path curvature is given by the reciprocal of the radius of the osculating circle. This is the circle that touches a curve on the concave side, and describes the curvature at any point on a curve [9].

At t_0, a dead reckoning update is transmitted. The velocity of the dead reckoning model is also given by *v*. At time t, the model position and actual position are given by P$_M$ $\left(\frac{1}{k}, vt\right)$ and P$_A$$\left(\frac{\cos(\theta)}{k}, \frac{\sin(\theta)}{k}\right)$, respectively.

As velocity is constant, the distance *l* travelled by both the modelled and the actual position in each time interval is equivalent. Using the standard arc length formula, the angle θ is found to be vkt. At each time step, the distance between the actual and the modelled position, d, can be calculated using Equation 2.

$$\begin{aligned} d &= \sqrt{\left(\mathrm{P_A}(x) - \mathrm{P_M}(x)\right)^2 + \left(\mathrm{P_A}(y) - \mathrm{P_M}(y)\right)^2} \\ &= \sqrt{\left(\frac{\cos(vkt)}{k} - \frac{1}{k}\right)^2 + \left(\frac{\sin(vkt)}{k} - vt\right)^2} \end{aligned} \qquad (2)$$

where

v is the velocity of the entity
k is the curvature of the path taken by the entity
t is the time

Both $\cos(vkt)$ and $\sin(vkt)$ can be approximated using the Taylor series expansion as follows:

$$\cos(vkt) \approx 1 - \frac{(vkt)^2}{2} + \frac{(vkt)^4}{4}$$

$$\sin(vkt) \approx vkt - \frac{(vkt)^3}{3}$$

Equation 2 can then be rewritten using these approximations, and used to calculate the time between updates, t_{update}, where δ is the spatial threshold distance; see Equation 3.

$$d = 0.5v^2kt^2 \Rightarrow t_{update} = \sqrt{\left(\frac{2\delta}{v^2k}\right)} \qquad (3)$$

According to Equation 1, absolute inconsistency can now be calculated as:

$$\int_{t_0}^{t_0+t_{update}} d(dt) = \int_{t_0}^{t_0+t_{update}} 0.5v^2kt^2 = \frac{1}{6}v^2kt^3 \Big]_{t_0}^{t_{update}} \qquad (4)$$

From Equations 3 and 4, it can be clearly seen that both spatial consistency and absolute consistency is related to curvature. Using these formulae, we will now examine the impact of the type of threshold used on both local inconsistency, when no network latency is present, and the generation of update packets.

3.1 Spatial and absolute consistency thresholds

A series of Matlab® simulations were performed for a constant velocity of 25 game units per second. The absolute consistency during a single period between the transmission of two packets was computed for various values of curvature and a spatial error threshold. A plot of curvature vs. absolute inconsistency is given in Figure 2 for various spatial error threshold values. A second set of simulations was performed using the absolute consistency error threshold. In this case spatial inconsistency was computed for each value of curvature. A plot of curvature vs spatial inconsistency is given in Figure 3 for various absolute consistency threshold values.

Figure 2 shows that absolute inconsistency decreases as path curvature increases when using a purely spatial threshold. When curvature is high the time between update packets is short and hence the absolute inconsistency value is low. In contrast, when curvature is low, the time between update packets is long and the absolute inconsistency is potentially unbounded. In contrast, Figure 3 shows that spatial inconsistency increases as path curvature increases when the absolute inconsistency threshold is employed. In this case, when curvature is high the entity diverges rapidly from the modelled path and it thus takes more time to exceed the chosen threshold value compared to the case of a low path curvature.

Figures 2 and 3, illustrate that the two threshold types are suited to opposite ends of the spectrum of curvature values. This supports the solution proposed in our previous work, which combined the spatial threshold and the absolute consistency threshold in the form of a novel hybrid threshold metric. In this case, both metrics are simultaneously evaluated. When one reaches the error threshold, an update packet is sent and both metrics are reset.

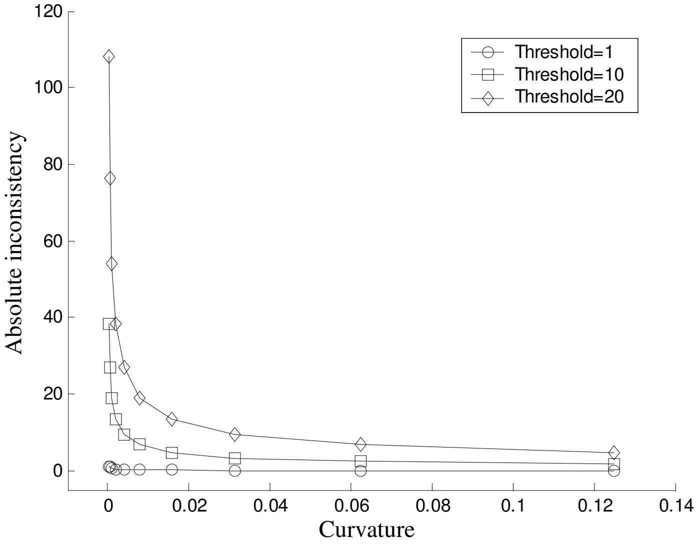


Figure 2 – Absolute consistency arising from use of a spatial threshold

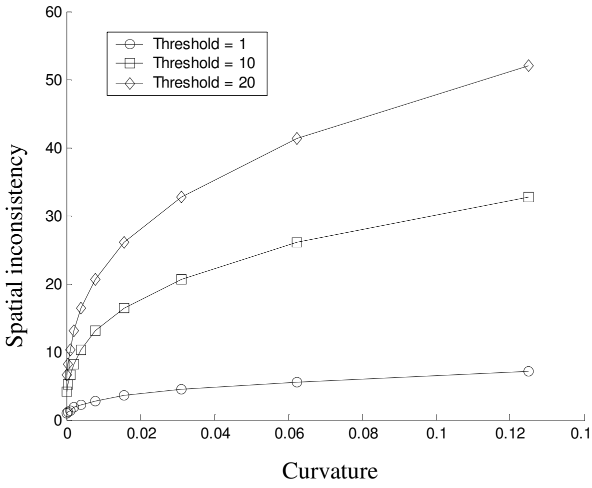


Figure 3 – Spatial inconsistency arising from use of an absolute consistency threshold

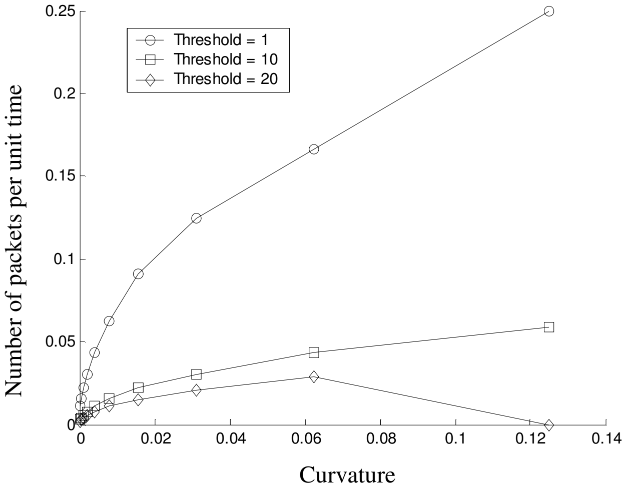


Figure 4 – Update packets required for varying path curvature and spatial threshold

3.2 Update packet generation

The number of update packets generated by the dead reckoning mechanism employed also impacts on consistency. Lower threshold values lead to increased numbers of update packets. This has the apparent effect of improving consistency. However the increase in packets may result in less efficient use of network resources, resulting in additional latency, and a possible reduction in consistency. It is therefore of interest to understand the effect of path curvature on the generation of update packets when the spatial and the absolute consistency thresholds are used. A number of Matlab® simulations were performed. Figure 4 is a plot of the average number of packets generated per unit time for various values of time in the case of a spatial threshold. Figure 5 is the same plot using the absolute consistency threshold. For each trial, path curvature and velocity were kept constant and the duration of each simulation was equal.

As expected, when the threshold increases, the number of packets decreases in both cases. However, the number of packets generated increases with curvature in both cases. The increase is more pronounced when the spatial threshold is employed because for a given threshold value and curvature, it is more likely that the spatial threshold will be exceeded than the absolute consistency threshold over a short time interval. In Figure 4, no update packets are generated when using the spatial threshold for some values of curvature. This occurs when the spatial threshold is larger than the radius of the circle traversed. With the absolute consistency threshold, such a situation cannot arise, as time always increases. Once again, this supports the use of a hybrid threshold scheme.

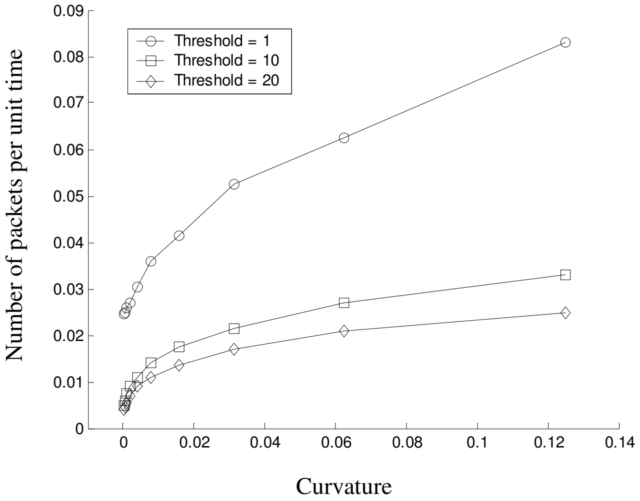


Figure 5 – Update packets required for varying path curvature and absolute consistency threshold

4. Experimentation and Analysis

The experimental test platform was implemented using the Torque engine. The engine was extended to include full logging, as well as the new dead reckoning threshold schemes (spatial, absolute consistency and the hybrid). The core networking architecture was modified. Traditionally, Torque implements a client-server distribution model. Dead reckoning, however, relies on the concept of replicated databases that are synchronised through updates. It was therefore necessary to extend Torque to support a distribution model based on peer-to-peer communication. This was implemented by treating a single user client server pair on the same machine as a peer in a multi-user session.

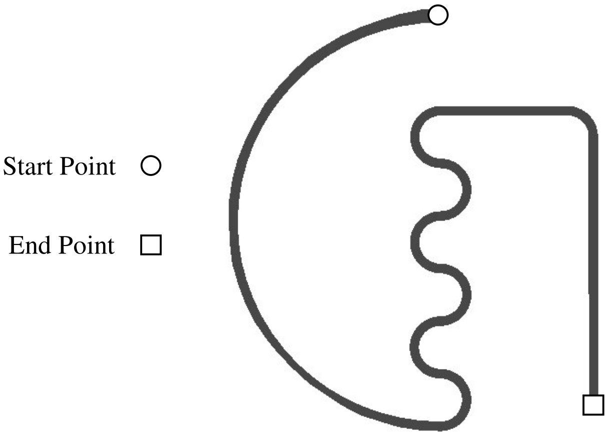


Figure 6 – Plan view of the environment course. The course was designed to incorporate and separate different aspects of likely path curvature.

Six participants of varying age, sex and gaming experience participated in the trials. Three participants were based in University of Salford, UK, and the others were based in NUI, Maynooth, Ireland. Each participant was first given a chance to familiarise themselves with the course, and the controls of the virtual environment. Usually this process lasted for the entirety of the course. A plan view of the course is given in Figure 6. The goal of the game is to be the first player to reach the finish line. The course was designed to incorporate and separate all aspects of path curvature that might be found mixed on a real race course. So that the effect of curvature during the simulation is examined, each section has been given a constant curvature.

Two participants were involved in each trial; one of whom was based in NUI, Maynooth, Ireland and the other in the University of Salford, UK. Six experiments were conducted during each trial. These experiments tested the use of the spatial threshold, the absolute consistency threshold, and the hybrid threshold. The clocks on participant machines were synchronized at start up using the Network Time Protocol (NTP)[10]. On each game tick (approximately 32 ms), each participant locally logged his or her own position along with the position of the remote client. Each participant also logged the number and type of updates transmitted during the simulation. Each log entry was time stamped with synchronised wall clock time. At the conclusion of the test, each participant completed a short questionnaire, which covered aspects of their experience in the virtual environment.

5. Results

In this section, we examine results collected in the live trials described in Section 4. These results differ from those presented in Section 3, as in this case, we are examining the inconsistency between the local client and the remote model, which will be affected by network latency [8]. Also, although client clocks were synchronised, the period of a client's tick depends on its performance and load (e.g. operating system tasks; background processes). Thus the tick period may vary with load and ticks are not synchronised across clients. These factors were kept in mind when performing comparisons.

We will first analyse the spatial and absolute consistency thresholds for a single user that is typical of all user data. Figure 7 details the absolute inconsistency measured when an absolute consistency threshold of 10 absolute consistency units is employed. This is the area under the curve of spatial difference against time. Figure 8 shows the spatial inconsistency arising from the use of a spatial threshold of 10 game units. As both figures present different measurements of inconsistency, they cannot be directly compared. However, as one might expect, both figures show that spatial and absolute inconsistency are bounded by the relevant threshold.

Figures 9 and 10 show the spatial inconsistency arising from the data presented in Figure 7 and the absolute inconsistency arising from Figure 8, respectively. A comparison of these shows that inconsistency in both cases is unbounded. By relating these plots to the curvature values of the navigated course, the impact of curvature becomes apparent. In the case of an absolute consistency threshold, Figure 9 shows that over the section of the track with a low, constant curvature, spatial inconsistency is lower than over the section of track with high, rapidly changing curvature. In contrast, when a spatial threshold is used, Figure 10 shows that over the section of the track with a low, constant curvature, absolute inconsistency is higher than over the section of track with high, rapidly changing curvature. The first update in both cases results in extreme consistency values, even though path curvature is not high. This occurs because the dead reckoning model has zero velocity, as both participants have just joined the environment. The actual position is moving away from the model position very quickly, mimicking the effect of very high path curvature. These results are interesting as they provide empirical evidence of the impact of curvature on the ability to maintain consistency using different threshold mechanisms.

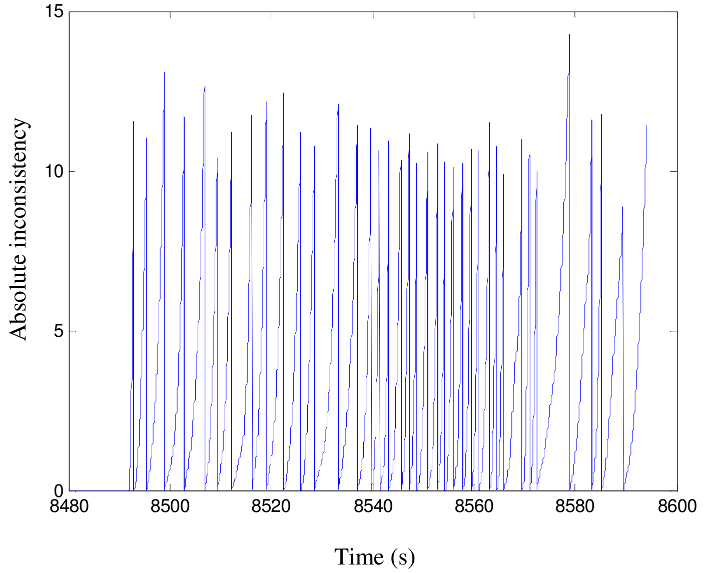


Figure 7 – Absolute inconsistency arising from use of absolute consistency threshold of 10

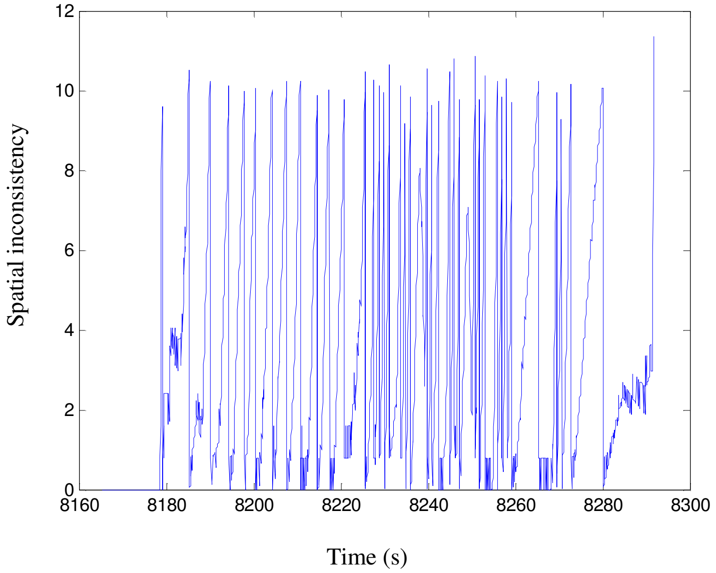


Figure 8 – Spatial inconsistency arising from use of spatial threshold of 10

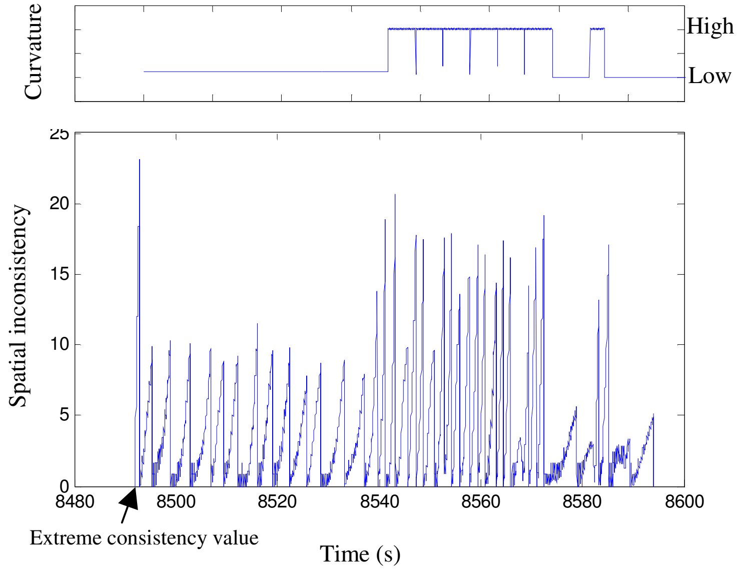


Figure 9 – The upper graph shows course curvature. The lower graph shows the spatial inconsistency arising from use of an absolute threshold of 10.

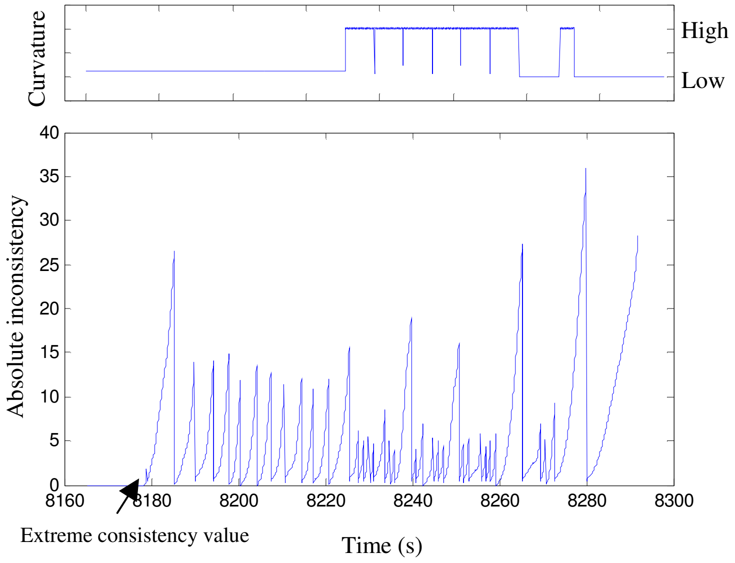
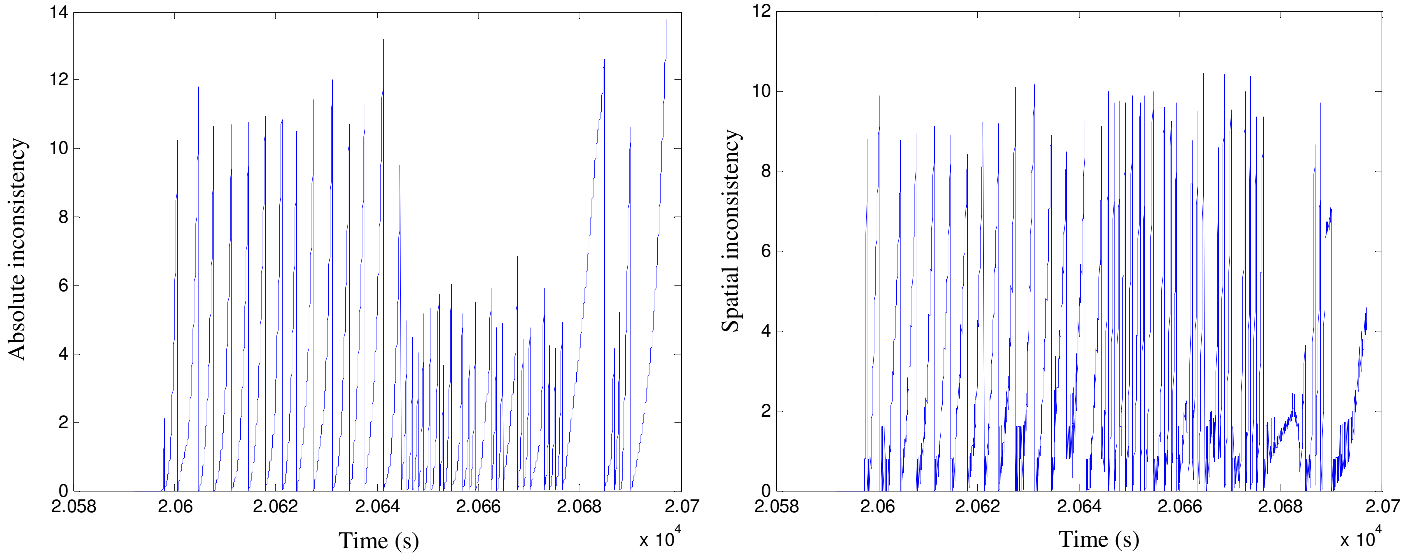


Figure 10 – The upper graph shows course curvature. The lower graph shows the absolute inconsistency arising from use of spatial threshold of 10.



(a) Absolute consistency

(a) Spatial inconsistency

Figure 11 – Absolute inconsistency and spatial inconsistency using a hybrid threshold of 10.

Figure 11 shows the measured inconsistency when the hybrid threshold is employed. We can see clearly from these results that both measures of inconsistency are now bounded. The effect of path curvature is evident, but has less impact, as one of the thresholds is always reached before inconsistency can become too large. This is further demonstrated in Figure 12. Here, the type of updates transmitted during a typical simulation when the hybrid threshold of 10 is employed is plotted. This is superimposed over the participant trajectory.

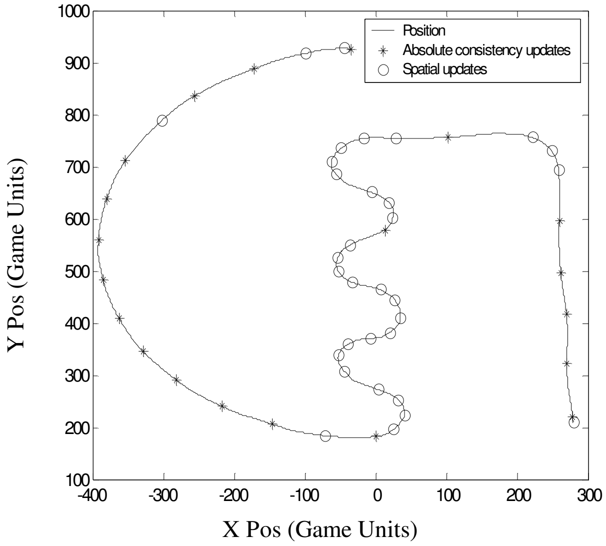


Figure 12 – Varying updates are transmitted as the curvature of the course changes

The type of updates transmitted is related to the curvature of the course. At the beginning of the simulation, when curvature is low, there is a higher frequency of breaches of the absolute consistency threshold. As curvature increases in the middle section, the spatial threshold is primarily reached. Towards the end of the course, the frequency of absolute consistency updates increases once again as the course curvature decreases. The short space between the first two updates can again be attributed to the initial model velocity of zero.

6. Conclusions

This paper has examined the impact of spatial and absolute consistency thresholds on the relationship between path curvature and inconsistency. The relationship was first expressed formally, and a link between curvature and both spatial and absolute inconsistency was established. The inconsistency arising from the use of both thresholds was then examined.

It was shown that each pure threshold method is sensitive to opposing curvature characteristics. When a spatial threshold is employed, low path curvature results in high absolute inconsistency. On the other hand, high spatial inconsistency is evident when path curvature is high, and an absolute consistency threshold is used. This proof strengthens the case for a recently proposed hybrid threshold, which incorporates both metrics simultaneously [4].

Trials were conducted over the Internet using the three threshold metrics. Results presented from this live situation further reinforce the theory and are consistent with those gained from simulation in our previous work.

Future work will examine an adaptable threshold that reacts to network usage and course curvature. In addition, the impact on qualitative aspects of game play will be investigated.

7. Acknowledgements

This work is supported by Science Foundation Ireland and Enterprise Ireland under grant IRCSET/SC/04/CS0289 and by Marie Curie HPMT-CT-2001-00390.

8. References

[1] Delaney, J. Latency Reduction in Distributed Interactive Applications using Strategy-based Models. Electronic Engineering, NUI, Maynooth, National University of Ireland, Maynooth**:** 2005,271.

[2] Gautier, L., C. Diot and J. Kurose "End-to-end Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet". *Proceedings of IEEE Infocom*,March, 1999.1470-1479.

[3] IEEE IEEE Standard for Distributed Interactive Simulation - Application Protocols IEEE Std 1278.1-1995. IEEE Std 1278.1-1995 (Revision of IEEE Std 1278-1993), IEEE**:** 1995.

[4] Roberts, D., D. Marshall, S. McLoone, D. Delaney, T. Ward and R. Aspin "Exploring the use of local inconsistency measures as thresholds for dead reckoning update packet generation". *DS-RT*, Montreal, Canada,10 12 October, 2005.

[5] Diot, C. and L. Gautier "A Distributed Architecture for Multiplayer Interactive Applications on the internet." *IEEE Network*, July/August,1999,6-15.

[6] Lui, J. C. S. "Constructing Communication Subgraphs and Deriving an Optimal Synchronization Interval for Distributed Virtual Environment Systems." *IEEE Transactions on Knowledge and Data Engineering*, September/October,2001,778-792.

[7] Correa, C., I. Marsic and X. Sun Semantic Consistency Optimization in Heterogeneous Virtual Environments, Technical Report CAIP-TR-267, Rutgers University, CAIP Center**:** 2002.

[8] Zhou, S., W. Cai, F. B. S. Lee and S. J. Turner "Time-space Consistency in Large Scale Distributed Virtual Environment." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, January 2004,2004,31 - 47.

[9] Anton, H. *Calculus with Analytic Geometry* New York, Chichester, Brisbane, Toronto, Singapore, John Wiley and Sons,(1995)

[10] Mills, D. L. "Internet Time Synchronization: The Network Time Protocol." *IEEE Transactions on Communications*, October,1991.