Exploring the Effect of Curvature on the Consistency of Dead Reckoned Paths for Different Error Threshold Metrics

Damien Marshall°, Séamus McLoone°, David Roberts*, Declan Delaney°, Tomás Ward°

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

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

dmarshall@eeng.nuim.ie,d.j.roberts@salford.ac.uk

Abstract

Dead reckoning is widely employed as an entity update packet reduction technique in Distributed Interactive Applications (DIAs). Such techniques reduce network bandwidth consumption and thus limit the effects of network latency on the consistency of networked simulations. A key component of the dead reckoning method is the underlying error threshold metric, as this directly determines when an entity update packet is to be sent between local and remote users. The most common metric is the spatial threshold, which is simply based on the distance between a local user's actual position and their predicted position. Other, recently proposed, metrics include the time-space threshold and the hybrid threshold, both of which are summarised within. This paper investigates the issue of user movement in relation to dead reckoning and each of the threshold metrics. In particular the relationship between the curvature of movement, the various threshold metrics and absolute consistency is studied. Experimental live trials across the Internet allow a comparative analysis of how users behave when different threshold metrics are used with varying degrees of curvature. The presented results provide justification for the use of a hybrid threshold approach when dead reckoning is employed in DIAs.

1. Introduction

Collaboration and competition are important factors of distributed interactive applications (DIAs). Despite the obvious conflict between them, they both require a certain level of consistency in order for the interaction to be useful and compelling. Consistency has been defined in many ways [1]. For the purposes of this paper, three types of consistency are considered, namely spatial, temporal and absolute consistency. A user's 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 latency. Network latency refers to the delay in transmission of packets and arises for a number of reasons. These include 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. The IEEE DIS standard dead reckoning mechanism [3] is a popular method that attempts to maintain a certain level of consistency while overcoming network delays and reducing network traffic for dynamic entities. Here, each participant transmits position and velocity data in an update packet whenever a model of their predicted movement and their actual movement differ by a pre-defined error threshold value. All other participants then use this data to model the behaviour of each entity. In the DIS standard the error threshold is based on the spatial distance between the two positions.

In previous work [9] we demonstrated that although the conventional approach of using a purely spatial error threshold for dead reckoning gives bounded spatial inconsistency, it nevertheless can lead to unbounded absolute inconsistency. A novel time-spatial threshold metric was proposed. This used local absolute inconsistency measures in determining when an update is required and was shown to result in bounded local absolute inconsistency but unacceptably large spatial inconsistency. A hybrid threshold combining both a spatial and a time-spatial threshold metric was subsequently proposed and this successfully resulted in both bounded spatial inconsistency and bounded absolute inconsistency.

In this paper we focus on the issue of user movement in relation to dead reckoning and each of the aforementioned threshold metrics. In particular the relationship between the curvature of movement, the different threshold metrics and absolute consistency is studied. A mathematical derivation of the relationship between path curvature and absolute consistency is determined, thus improving our understanding of the issue in question. Experimental live trials across the Internet are conducted using an industry standard games engine called Torque[4]. The obtained results clearly illustrate the effects of curvature on the various error threshold metrics used with the dead reckoning entity update mechanism. They also provide further justification for the use of a hybrid threshold approach when dead reckoning is employed in DIAs.

The rest of the paper is structured as follows. For the convenience of the reader, the various metrics used for measuring consistency are outlined in the next section. In Section 3, a mathematical derivation of the relationship between path curvature and absolute consistency is determined. Simple simulations illustrating this relationship are also outlined. Section 4 provides an overview of the internet trials conducted to investigate the effects of path curvature on consistency for different threshold metrics. The results of these trials are presented and analysed in section 5. The paper ends with some conclusions and suggestions for future work in section 6.

2. Background Information

Different measures of consistency have been proposed by several authors. Diot and Gautier 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. 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 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]. In their paper, they demonstrate how a two-dimensional and three-dimensional view of the same object can be semantically consistent because the meaning that users draw from both views is the same. 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 time-space inconsistency arising from a Distributed Interactive Application simulation before execution time. The inconsistency metric is given in Equation 1 below:

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

Here, $\Delta(t)$ is the difference between the position of a local object and its remote replication over a duration t, t_o is the start time of the inconsistency, τ is the duration and ε is the minimum perceivable distance.

In previous work it was demonstrated, using Zhou's metric, that local absolute inconsistency is unbounded when a spatial threshold is employed with the dead reckoning entity update mechanism [9]. To rectify this, a new dead reckoning time-space threshold metric was proposed. This metric employed Zhou's measure of local absolute inconsistency in determining when a dead reckoning update should be transmitted. Simply stated, the time-space threshold is the area under the graph of the difference between the actual and predicted paths over time. Such an approach is important as it takes into account temporal aspects, such as the duration of the error, as well as spatial error in determining when to send an update.

It was subsequently shown that the use of the time-space threshold metric placed a bound on local absolute inconsistency but could, unfortunately, result in unacceptably large local spatial inconsistency. A hybrid solution was developed that simply employed both the spatial and the time-space threshold metrics. This metric is referred to as the hybrid error threshold metric and simulation studies showed that it guaranteed a bound on both local spatial inconsistency and local absolute inconsistency. The reader is directed to [9] for a detailed analysis of this work.

The remainder of this paper investigates the effect of path curvature on the dead reckoning mechanism, with particular consideration given to the aforementioned threshold metrics.

3. Path Curvature and Consistency

A mathematical derivation of the relationship between the path curvature and absolute consistency is now outlined.

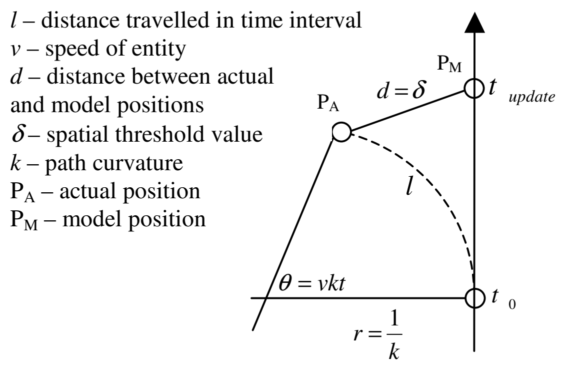


Figure 1. Actual and dead reckoning model positions

Consider the case where the curvature, *k*, and speed, *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 [10]. At t_0 a dead reckoning update is transmitted. The speed of the dead reckoning model is also given by *v*. At time *t*, the model position and actual position are given by P_M and P_A respectively:

$$P_M\left(\frac{1}{k}, vt\right) \quad \text{and } P_A\left(\frac{\cos(\theta)}{k}, \frac{\sin(\theta)}{k}\right) \qquad (2)$$

As speed 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 as follows:

$$\begin{aligned} d &= \sqrt{(P_A(x) - P_M(x))^2 + (P_A(y) - P_M(y))^2} \\ &= \sqrt{\left(\frac{\cos(vkt)}{k} - \frac{1}{k}\right)^2 + \left(\frac{\sin(vkt)}{k} - vt\right)^2} \end{aligned} \qquad (3)$$

Approximating cos(*vkt*) and sin(*vkt*) using the Taylor series expansion as follows:

$$\cos(vkt) \approx 1 - \frac{(vkt)^2}{2} + \frac{(vkt)^4}{4}, \sin(vkt) \approx vkt - \frac{(vkt)^3}{3} \quad (4)$$

reduces Equation 3 to:

$$d = 0.5v^2kt^2 \qquad (5)$$

Hence, the time between updates, t_{update} , can be expressed as:

$$t_{update} = \sqrt{\left(\frac{2\delta}{v^2k}\right)} \qquad (6)$$

where δ is the spatial threshold distance. According to Equation 1, absolute inconsistency can now be calculated as:

$$\int_{t_0}^{t_0+t_{update}} d(dt) = \int_{t_0}^{t_0+t_{update}} 0.5v^2kt^2 = \frac{1}{6}v^2kt^3\Big]_{t_0}^{t_{update}} \qquad (7)$$

From Equations 5 and 7 it can be clearly seen that both spatial consistency and absolute consistency is related to curvature *k*. These formulae are simulated in the following subsections in order to fully understand the impact of curvature on inconsistency for different threshold metrics. In these simulations, network latency is set to zero. Since the number of entity update packets also affects consistency, its relationship with respect to curvature and the different threshold metrics is also considered.

3.1 Spatial and Time-Space Threshold Metrics

Here, speed *v* is arbitrarily set to a constant value of 25 game units per second. Figure 2 shows a plot of absolute inconsistency for various values of curvature, where the underlying dead reckoning method employs a spatial error threshold of 1, 10 and 20 game units respectively. Figure 3, on the other hand, presents a plot of spatial inconsistency for varying curvature, when a time-space threshold metric is used. Threshold values of 1, 10 and 20 game units are considered. The inconsistency values obtained (either absolute or spatial) relate to a single period between the transmission of two consecutive entity update packets.

Figure 2 shows that absolute inconsistency decreases as path curvature increases when a spatial threshold metric is employed. Note that absolute consistency is a function of both time and distance. Therefore, for a constant distance the inconsistency value is directly proportional to the length of time. Hence, when curvature is high the time between update packets is short and the absolute inconsistency value is low. On the other hand, when curvature is low, the time between update packets is long and the absolute inconsistency is potentially unacceptably large.

In contrast, Figure 3 shows that spatial inconsistency increases as path curvature increases when the time-space threshold metric is employed. In this case, when curvature is high the entity diverges rapidly from the modelled path in a short time period, whereas with low curvature the entity diverges less

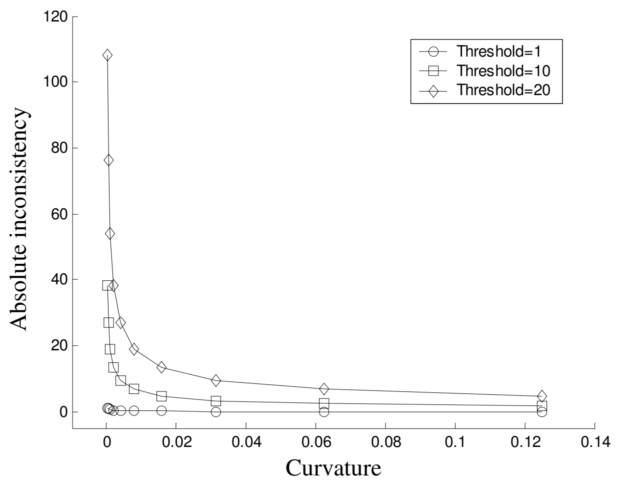


Figure 2. Absolute inconsistency for varying curvature - spatial threshold metric used

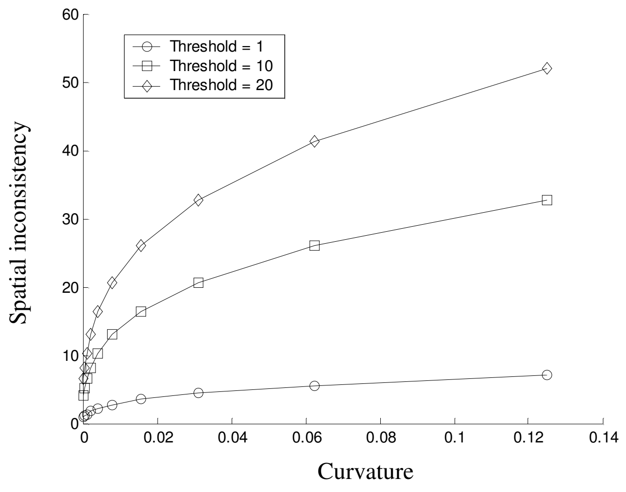


Figure 3. Spatial inconsistency for varying curvature - time-space threshold metric used

rapidly over a longer period of time. Note, in both cases, the time-space threshold value is the same, but reached at two different times. Consequently, in the case where the time period is longer the spatial inconsistency is smaller and when the time period is short, the spatial inconsistency is very large.

These results clearly 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 [9], which combined the spatial threshold and the absolute inconsistency threshold in the form of a 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. Hence, both spatial and absolute inconsistencies remain within acceptable bounds.

3.2 Update Packet Generation

As already stated, the number of update packets generated by the dead reckoning mechanism also impacts on consistency. Lower threshold values leads to better model prediction and this has the apparent effect of improving consistency. However, lower threshold values also increase the number of update packets which, in turn, may result in less efficient use of network resources, resulting in additional latency and a subsequent reduction in consistency. It is therefore important to understand the effect of path curvature on the generation of update packets in relation to the use of the spatial and the time-space threshold metrics. To do this, a number of Matlab simulations were conducted. Equation 2 was used to govern the position of the actual and dead reckoned path. The speed of the simulated entity was set to 25 game units per second. The simulated entity moved at a constant curvature for the entirety of the simulation. Each simulation lasted for a period of 60 seconds. The number of packets generated was recorded at the end of each simulation. Figure 4 shows the number of update packets required for differing curvature values when a spatial threshold metric is used. Figure 5 shows a similar plot for the case when a time-space threshold metric is employed.

Comparing both figures to their counterparts in Figures 2 and 3, the trade off between throughput and consistency is clearly seen. For lower thresholds, inconsistency is low. However, this has the effect of requiring a large number of packets. The impact of curvature on both threshold metrics is also evident from both figures. As curvature increases, the number of packets generated also increases in both cases. Intuitively, we would expect that using a spatial threshold metric results in an increase in packets relative to the time-space threshold at higher curvatures, and that employing a time-space threshold results in an increase in packets at lower curvatures relative to the spatial threshold. While the results confirm this, we cannot carry out a truly accurate comparison, as there is no established link between the metrics. These results reinforce the data presented in Figures 2 and 3, as they show that the performance of both thresholds is sensitive to opposing curvature values, and support the need for a hybrid scheme. It is also interesting to note that 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 time-space threshold, such a situation cannot arise, as time always increases, meaning that the threshold will be breached at some

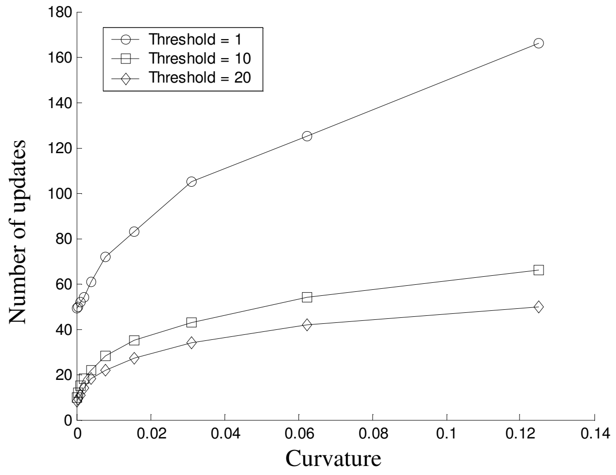


Figure 4: Update packets required for varying path curvature - spatial threshold metric used

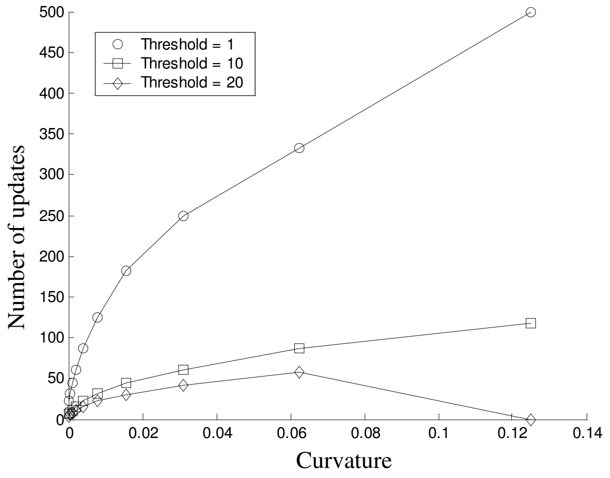


Figure 5: Update packets required for varying path curvature – time-space threshold metric used

stage during the simulation. Once again, this supports the use of a hybrid threshold scheme.

4. Experimentation – Internet Based Trials

To strengthen our mathematical analysis in the previous section, a number of live Internet-based trials were conducted, using an experimental test platform implemented in the Torque games engine. The engine was extended to include full logging, as well as the new dead reckoning threshold schemes (spatial, time-space and the hybrid). A novel extension to the core networking architecture was also implemented. Traditionally, Torque implements a client-server distribution model, with the server governing the final state of the simulation for each client. However dead reckoning relies on the concept of replicated databases that are synchronised through updates, and also requires that each client drives their own version of the simulation. It was therefore necessary to extend Torque to support a distribution model based on peer-to-peer communication. This was achieved by treating a single user client server pair on the same machine as a peer in a multi-user session. It operates as follows. When a client enters the environment, it joins its own local server. It then queries a master server to obtain a list of all servers running the same environment, and proceeds to join each server. During the execution of the environment, each client transmits any required updates to every server involved in the simulation. However, for each client, only the single local server governs the final local state of the virtual environment.

Six participants of varying age, sex and gaming experience participated in the trials, which took place across the Internet, ensuring that data was subject to real network problems such as latency and jitter. Three participants were based in University of Salford, UK, and the others were based in the National University of Ireland, Maynooth, Ireland. Each participant was first given a chance to familiarise themselves with the course and the controls of the virtual environment. A plan view of the course is given in Figure 6.

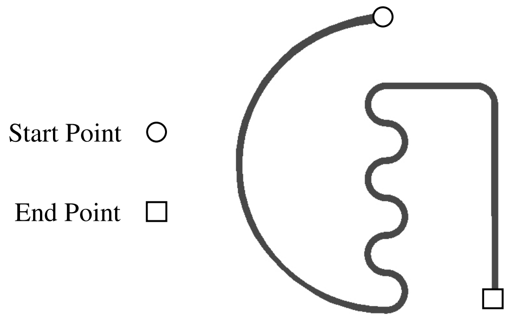


Figure 6. Plan view of the environment course.

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. For ease of analysis, each section of the course has been given a constant curvature.

Two participants, one from each of the two universities, were involved in each trial. Six experiments were conducted during each trial. These experiments tested the use of the three different threshold metrics. The clocks on participant machines were synchronized at start up using the Network Time Protocol [11]. On each game tick (approximately 32 ms) each participant logged his/her own position, along with the position of the remote client. Each participant

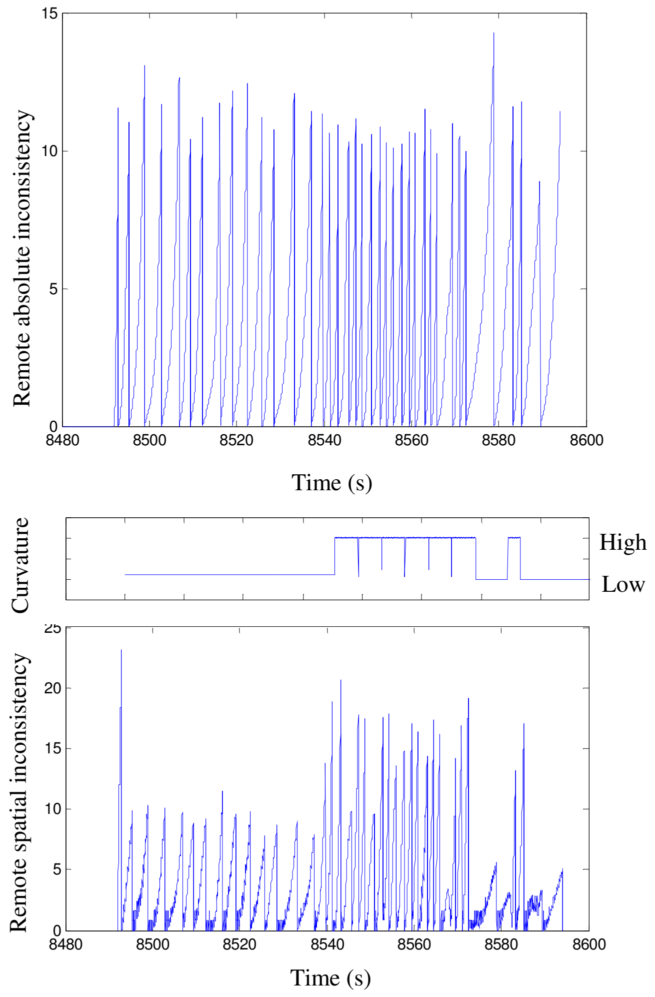


Figure 7. Upper graph shows remote absolute inconsistency; middle graphs shows course curvature; lower graph shows the remote spatial inconsistency. A time-space threshold of 10 was employed.

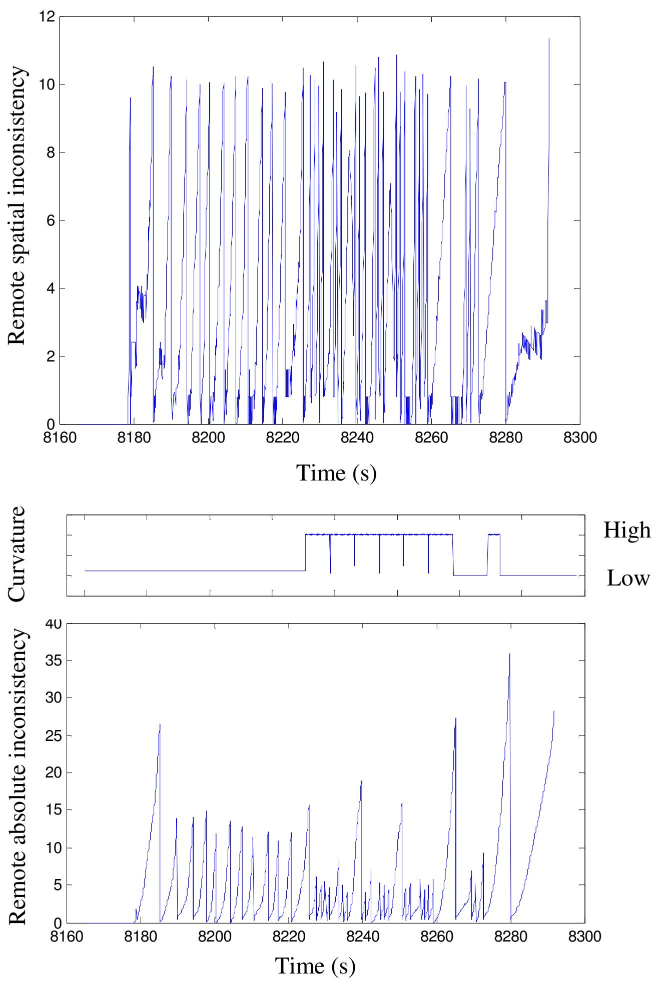


Figure 8. Upper graph shows remote spatial inconsistency; middle graphs shows course curvature; lower graph shows the remote absolute inconsistency. A spatial threshold of 10 was employed.

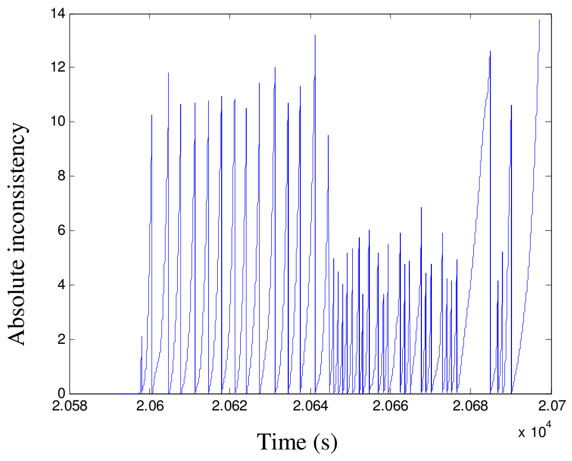
also logged the number and type of updates transmitted during the simulation. Each log entry was time stamped with synchronised wall clock time.

5. Results and analysis

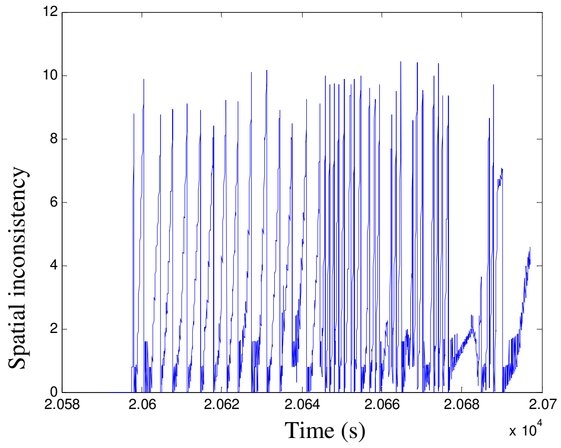
We now examine the results collected in the live trials described in Section 4 above. These results differ from those presented in Section 3 as, in this case, we are analyzing the inconsistency between the local client and the remote model. Hence the data is clearly affected by actual network latency and jitter. Also, although client clocks were synchronised, the period of a client's tick depends on its performance and load (e.g. operating system tasks). Thus, the tick period may vary with load and ticks are not synchronised acrossed clients. These factors were kept in mind when performing comparisons.

Figure 7 shows plots of remote absolute and spatial inconsistency obtained when a time-space threshold of 10 game units seconds is used. Figure 8, on the other hand, shows similar plots for a spatial threshold of 10 game units. A plot of the course curvature is also shown in both figures in order to aid analysis. It should be noted that, while the results presented in this section are for a single user, they nevertheless are typical of all user data. Further more, the actual value of time in the graphs of both figures is not important. These simply represent the point in time in which data was collected with respect to the start of the time server.

The upper plot in Figure 7 clearly shows that absolute consistency is bounded at 10 units (allowing for latency and jitter) for a time-space threshold of 10 units, irrespective of path curvature. However, the



(a) Time-space inconsistency



(b) Spatial inconsistency

Figure 9. Absolute and spatial inconsistency using a hybrid threshold of 10.

bottom plot shows that this is clearly not the case for remote spatial inconsistency. Comparing this plot with the curvature values of the navigated course clearly highlights the impact of curvature. Over the section of the course with a low curvature, spatial inconsistency is lower than over the section of track with high curvature.

In contrast, the upper plot in Figure 8 shows that it is spatial consistency that is bounded at 10 units (once again allowing for latency and jitter) when a spatial threshold of 10 units is used, irrespective of path curvature. The bottom plot shows that it is affected by curvature in this case. Once again, comparing this plot with the curvature values of the navigated course highlights the impact of curvature. Over the section of the course with a high curvature, absolute inconsistency is lower than over the section of track with low curvature. These results provide excellent empirical evidence of the impact of curvature on the ability to maintain consistency using different threshold mechanisms.

Logically, combining both threshold metrics should bound inconsistency irrespective of the value of curvature. Figure 9 shows the absolute and spatial inconsistency measured when the hybrid of both metrics is utilised. We can see clearly from these results that both measures of inconsistency are indeed bounded at a value of 10, allowing for latency and jitter. The effect of path curvature is still evident but no longer prevents bounding of the inconsistency value, since one of the thresholds is always reached before inconsistency can become too large. This latter fact isfurther illustrated in Figure 10. 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.

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 time-space threshold. As curvature increases in the middle section, the spatial threshold is primarily reached. Towards the end of the course, the frequency of time-space updates increases once again as the course curvature decreases.

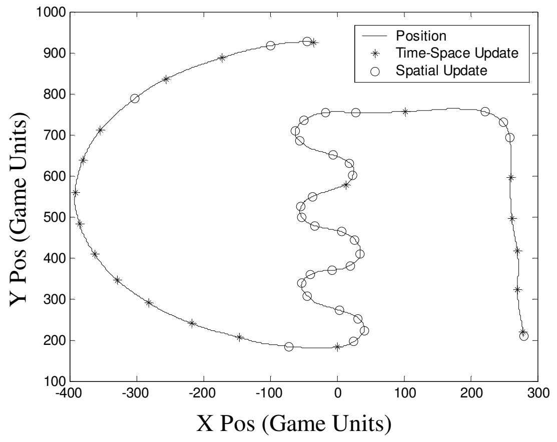


Figure 10. Varying updates are transmitted as the curvature of the course changes

Finally, for the sake of completeness, the average number of packets generated per person for the different threshold metrics is displayed in Figure 11 below. Threshold values of 5 and 10 game units are considered. As expected, the results confirm that the hybrid threshold results in an increase in packets across the network. This increase in packet number is necessary to ensure better consistency for the application. Ironically, however, this increase will possibly reduce consistency at times when the network bandwidth has high usage, due to an increase in network traffic and, therefore, an increase in network latency. The actual overall impact on consistency is therefore not certain and this remains an issue for future studies.

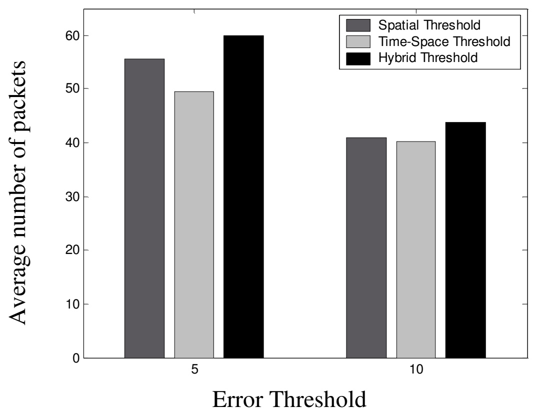


Figure 11. Average number of packets generated for the different error threshold metrics during the live Internet trials.

9. Conclusions

This paper has examined the impact of spatial and time-space error threshold metrics on the relationship between path curvature and consistency. The relationship was first expressed formally, and a link between curvature and both spatial and time-space curvature was established. The inconsistency arising from the use of both thresholds was then examined. Both mathematical simulations and actual experiments, conducted across the Internet, clearly showed that each individual threshold metric is sensitive to opposite ends of the curvature spectrum. When a spatial threshold is employed, low path curvature results in high absolute inconsistency. Furthermore, when a time-space threshold is used, high path curvature gives high spatial inconsistency. This proof strengthens the case for a recently proposed hybrid threshold, which incorporates both metrics simultaneously [9].

Unfortunately, the hybrid threshold approach results in an increase of update packets across the network. Therefore, depending on the capacity of the network and its loading, it has a possible adverse affect on consistency. The overall impact on consistency requires future detailed study, which is beyond the scope of this paper.

In this work, the same threshold value was employed for both spatial and time-space threshold within the hybrid threshold approach. However, this may not be appropriate. Future work will explore means of comparing spatial and time-spatial threshold metrics, so that their performance can be directly compared, and so that suitable threshold values for use in the hybrid threshold scheme can be chosen.

10. Acknowledgements

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

11. References

[1] Delaney, D., T. Ward and S. McLoone, "On Consistency and Network Latency in Distributed Interactive Applications: A Survey - Part 1". *Presence:Teleoperators and Virtual Environments*, Vol. 15, No. 2 April 2006

[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, pp. 1470-1479.

[3] IEEE. *IEEE Standard for Distributed Interactive Simulation - Application Protocols IEEE Std 1278.1-1995* IEEE. (1995)

[4] Marshall, D., A. McCoy, D. Delaney, S. McLoone and T. Ward "A Realistic Distributed Interactive Application Testbed for Static and Dynamic Entity State Data Acquisition". *Irish Systems and Signals Conference*, Belfast, Ireland, June 2004, pp. 83 -88.

[5] Diot, C. and L. Gautier. "A Distributed Architecture for Multiplayer Interactive Applications on the internet." *IEEE Network* 13(4), 1999,pp. 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* 13(5), 2001, pp. 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)* 14(1). January 2004, pp. 31 - 47

[9] 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". *Distributed Simulation and Real Time Applications*, Montreal, Canada, October 2005, pp.195-202

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

[11] Mills, D. L. "Internet Time Synchronization: The Network Time Protocol." IEEE Transactions on Communications *39(10),1991*