

Using Perceptual Feedback To Determine Time-Space Threshold Values For Dead Reckoning

Alan Kenny, Séamus McLoone, Tomás Ward and Declan Delaney

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

E-mail: akenny; seamus.mcloone; tomas.ward @eeng.nuim.ie

ABSTRACT

Entity state prediction mechanisms are used in order to reduce the number of packets required to maintain a consistent state in a Distributed Interactive Application (DIA). Typically in the case where the entity is representing a participant in a networked game this is achieved by continually comparing the output of a prediction algorithm against a player's actual state. The state usually comprises position and orientation information in such cases. If the error exceeds a pre-defined threshold value, then an update packet is transmitted, which contains the player's latest trajectory information. However, obtaining a suitable threshold value remains one of the key challenges that face such entity state prediction techniques. Furthermore, these methods can employ two different threshold metrics. These are *spatial*, which exploits distance measures, and *time space*, which uses both time and distance measures. While a spatial threshold value can be arguably determined based on a prior knowledge of the gaming environment, it remains difficult, at best, to obtain a corresponding value for the time space threshold metric.

This paper proposes the novel use of user perception as a suitable means to solve the aforementioned problem. Here we employ the most common entity update mechanism, namely *dead reckoning*, and use perceptual feedback to determine suitable threshold values for both spatial and time space threshold metrics. This involves collecting linguistic feedback on short scenes recorded from a racing game. This technique is compared and contrasted with an alternative method whereby equivalent spatial and time space threshold values are obtained based on a common measure of inconsistency. Details of the experimentation and an analysis of the results are presented within.

Keywords – Distributed Interactive Applications (DIAs), Psycho-Perceptual Measures, Dead

Reckoning, Spatial Error Threshold, Time Space Error Threshold

1. INTRODUCTION

Distributed Interactive Applications (DIAs) have to continually deal with the limitations of their underlying networks, i.e. latency, jitter and network congestion. As a result, it is desirable to minimise the number of packets that must be sent across a network in order to reduce the possibility of remote users having an inconsistent view. To achieve this, prediction mechanisms are used to model remote entities.

One of the most popular techniques used to date is the entity state prediction mechanism known as dead reckoning. This was introduced in the IEEE Distributed Interactive Simulation (DIS) standard [1] and has become the standard for commercial games, such as Doom, Quake and Tribes II. Dead reckoning is a method of predicting a user's future actions based on their dynamics, which results in the transmission of less data to remote nodes. The most basic of these is to set the new position and velocity to the transmitted position and velocity, which is known as first order dead reckoning and is employed in this paper. Further information on dead reckoning can be found in [1,2]. Other packet reduction techniques used in DIAs include the area of interest management, Hybrid Strategy Model, data compression and dynamic load balancing [3-6].

One of the key factors in all entity update methods is when to send the updated information. For a dead reckoning model an update is sent once a certain tolerance value has been exceeded. This value is known as the error threshold. Typically this threshold value is arbitrarily chosen and generally reflects what 'appears' to be appropriate with respect to the underlying application.

Traditionally the most popular error metric has been spatial distance. Spatial distance compares the distance between a player's actual position and their local model. If the distance exceeds the error threshold value then an update is sent. The spatial metric is popular as it is simple to implement and the game environment can be used as a reference

point to determine suitable error thresholds. For example, a narrow racing track may require a tighter threshold than a wide track, as an entity leaving the track would be apparent sooner on the narrow track.

However, the spatial metric does not take the duration of an error into consideration, which led to the development of the time space metric [7]. For example, if the model continually has an inaccuracy just below the error threshold the spatial metric will allow this error to continue indefinitely. As the time space metric takes the duration of an error into consideration it will eventually send an update packet to correct this scenario, resulting in greater overall consistency.

Time space error refers to the cumulative spatial error over time. Figure 1 shows the time space error over one time period, where D represents the spatial distance error. Effectively this is the area under the curve between the local player's actual position and the local model over time. Similar to the spatial metric, the cumulative error is then compared to an error threshold. The biggest problem with the time space metric is determining a suitable error threshold. Unlike the spatial metric the environment alone cannot be used as a reference point for potential thresholds.

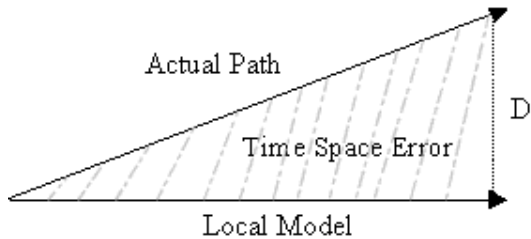


Figure 1: Time Space Metric Error

This paper proposes the novel use of perceptual feedback to determine equivalent spatial and time space error thresholds. We describe an experiment that collects relevant user perceptual feedback. From the collected perceptual feedback a lookup table is generated, for equivalent spatial and time space thresholds. Recent research has proposed a method of determining equivalent threshold values based on a common measure of inconsistency, namely the mean squared error [8]. A similar lookup table is generated using the mean squared error. The results for the two methods are compared and some possible outcomes are discussed.

The remainder of the paper is structured as follows. Section 2 details the design and implementation of the experiment used to collect information pertaining to the end-user perceptual experience. The resulting data is then analysed and discussed in section 3. Finally some conclusions and suggestions for future work are given in section 4.

2. EXPERIMENTATION

This section details the design and implementation of the experiment used to gather user perceptual feedback. The primary design goal was to examine the performance of the dead reckoning model for both spatial distance and time space error metrics.

2.1 Video Clips

In order to obtain the required feedback, a set of game-like video clips was created. This was achieved by recording the movements of a computer controlled entity or 'bot' under various conditions. The first scenario was created under ideal conditions, with no error or latency, and was used as the benchmark video. Each subsequent video consisted of the bot modelled with dead reckoning under various error thresholds. The user was then asked to compare the models with the original benchmark video. Feedback from our previous work [9] suggested that subjects find it difficult to continually recall the benchmark video. As a result the benchmark video was repeated after every four model videos. The latency was set to 200ms, with random jitter set to between $\pm 10\%$ of the latency value. These latency and jitter values were chosen as it has been shown that the average transmission times fall within the region of 200ms [10].

One of the main challenges encountered in designing this experiment was the creation of a suitable track for the bot to race around. In order to avoid subject fatigue during the experiment, due to long video durations, the track had to be relatively short. This resulted in an elliptical course being chosen. The Torque Game Engine [11] was used to create the track and scenarios used in this experiment. Most of the game measurements, such as the spatial distance, are calculated in Torque Game Units (tgu). As a reference point the track used in this experiment is approximately 100 tgu in width, 250 tgu in length and the two straight sections are about 30 tgu wide. Finally, the various games scenes were recorded as AVI files using FRAPS (<http://www.fraps.com>), a utility designed for recording game footage.

2.2 User Feedback

Previously our work in [9] highlighted bot smoothness, or 'jumpiness', as being a challenge in rating a video clip. A bot may be very accurate but nevertheless appear to 'jump' from time to time, which resulted in lower ratings. As a result users were asked to rate the model using two measures, its smoothness and motion accuracy. It was hoped that this would result in more realistic results for the motion accuracy score.

Linguistic variables were used to obtain subject feedback. The player smoothness and motion accuracy variables were rated as *Extremely Poor*, *Very Poor*, *Poor*, *Okay*, *Good*, *Very Good* and *Excellent*. A seven-point linguistic scale was chosen in order to avoid difficulties in quantifying specific levels of accuracy and it has been shown that humans can reliably distinguish between seven distinct states [12].

Subject feedback from our previous experiment indicated that staying focused for the duration of an experiment was difficult. In order to avoid this, the duration of the experiment was kept under fifteen minutes and subjects were asked to give continuous feedback about the overall quality of the scene, both player smoothness and motion accuracy. The continuous feedback was recorded on a discrete sliding scale from *Poor* to *Good*. A smaller scale was chosen for the continuous feedback in order to make it quick and easy for subjects to update their score. This also gave a greater level of interaction in the experiment, which more closely resembles the experiment source material, and would potentially allow for closer examining of a scene to determine when and for how long a subject's perception was altered. The application used in this experiment is shown in Figure 2.

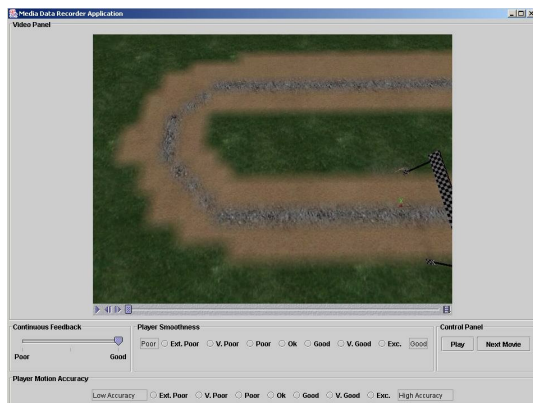


Figure 2: Java Media Data Recorder playing the Elliptical Racing track

2.3 Experimental Set-up

Before the experiment began each subject had the task explained to them and were shown a demonstration video to illustrate the tasks involved in the experiment. In order to avoid biasing the results, the difference between the two error metrics was not explained to the subjects. The tests began once the subject felt confident they understood the requirements for the experiment. The experiment consisted of the subject watching a video clip once, whilst giving continuous feedback. In addition, at the end of each video, participants were asked to rate Player Smoothness and Motion Accuracy.

There was a total of twenty-eight video clips per experiment. A demonstration video was also shown to participants to ensure that they understood the experiment. Six of the experiment videos were benchmark videos, twenty were unique models and two were duplicates. Each video lasted approximately twenty seconds and the entire experiment lasted approximately fifteen minutes.

3. ANALYSIS AND DISCUSSION

A total of ten subjects took part in this experiment, consisting of six males and four females, ranging in age from fifteen to thirty five. All subjects had some level of experience with using a computer, while six had some experience with computer games and four had experience with networked games.

The spatial thresholds used in the experiment were 6, 7.5, 9 and 10.5 tgu. These thresholds were chosen as our previous work indicated that the perceptual ratings would fall below acceptable within this region. A more explorative set of time space error thresholds were chosen, specifically 1, 2, 3, 4, 5 and 7 tgu.

3.1 Experimental Results for Discrete Feedback

This section is representative of the data collected from the subjects' ratings after viewing a video and does not take continuous feedback into consideration. Figure 3 represents the perceptual rating for the motion accuracy. For the four spatial thresholds, 6, 7.5, 9 and 10.5 tgu, the perceptual rating is extracted from the graph in Figure 3a. These perceptual ratings are then applied to the motion accuracy scores for the time space metric in Figure 3b. For example, Figure 3a highlights the perceptual rating for the spatial threshold of 7.5 tgu, which has a rating just above *ok*. The equivalent perceptual rating in Figure 3b gives a time space error of 2.35 tgu. From this information a perceptual lookup table can be generated that relates equivalent spatial and time space thresholds, which can be seen in Table 1.

It should be noted that the data for time space graph is not entirely smooth. This is most likely a result of the relatively low number of participants and should be taken into consideration when analysing these results. Increasing the number of participants should ameliorate this issue.

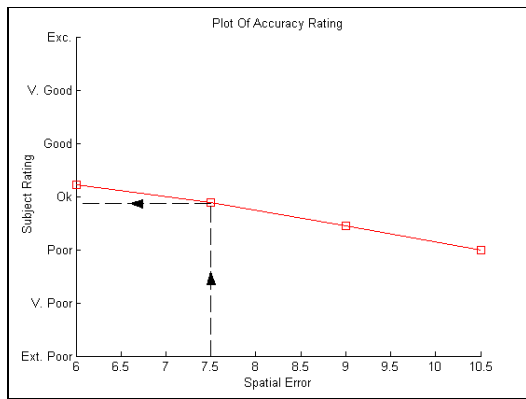
For comparison purposes, a lookup table is also determined according to [8]. The mean squared error is calculated by summing the absolute spatial inconsistency at every time interval and taking its average over the total time. Larger error thresholds will naturally result in larger mean squared errors. Figure 4a shows the mean squared error for the spatial metric. The mean squared error for each of

the four threshold values is highlighted. Figure 4b highlights the mean squared error for the time space threshold. By taking the corresponding spatial mean squared error and plotting it on this graph, equivalent spatial and time space thresholds can be determined and a lookup table generated. Table 2 represents the lookup table generated from Figure 4 and excludes the spatial threshold of 10.5 as it goes beyond the region covered in Figure 4b.

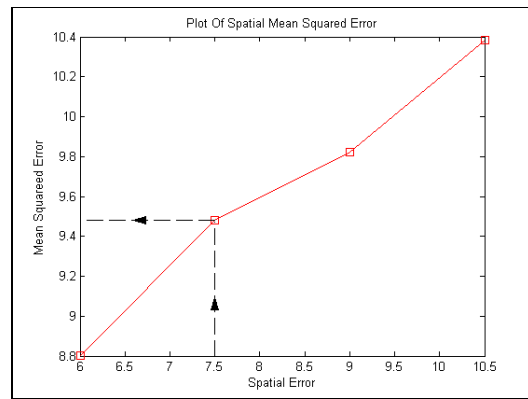
Interestingly only the smallest equivalent threshold in Table 1 corresponds to it's equivalent in Table 2, with values of 1.6 and 1.8 tgu respectively for the spatial threshold of 6 tgu. It is possible that for above average perceptual ratings the lookup tables will match. In this case both the spatial and time space graphs are likely to exhibit a steeper

decline in perceptual rating, from *excellent* to just above *ok*, for a spatial threshold from 0 to 6 tgu.

However, for large error thresholds there does not appear to be a correlation between the two lookup tables. For example, a spatial error of 7.5 gives a time space error of 2.35 in Table 1 and 4.1 in Table 2, which is a reasonable difference. It should be noted that the perceptual rating has a defined scale; from *excellent* to *extremely poor*, while the mean square error does not and will continue to grow as the error threshold increases. In other words, no matter how large the error threshold gets the perceptual feedback would be limited to *extremely poor* while the mean squared error would become very large.



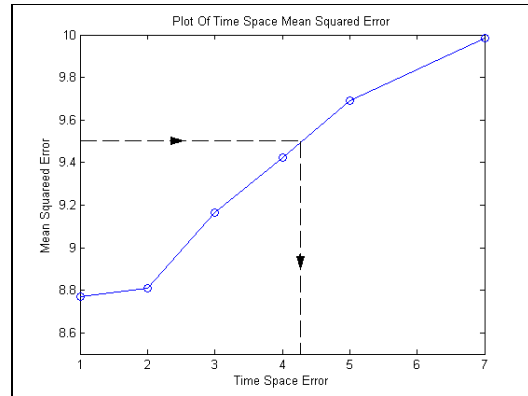
(a)



(a)



(b)



(b)

Figure 3: Plot of the Motion Accuracy Rating for the (a) Spatial and (b) Time Space metrics

Figure 4: Plot of the Mean Squared Error for the (a) Spatial and (b) Time Space metrics

Spatial	Perceptual Rating	Time Space
6	Just Above Ok	1.6
7.5	Just Below Ok	2.35
9	Between Ok/Poor	2.85
10.5	Poor	4.5

Table 1: Lookup table for Perceptual Ratings

Spatial	Mean Squared Error	Time Space
6	8.8	1.8
7.5	9.5	4.1
9	9.8	5.9

Table 2: Lookup table for Mean Squared Error

Ultimately it is the region that garners above acceptable error thresholds that is of use to developers. The mean squared error can theoretically generate equivalent time space values for very large spatial thresholds, but they would result in poor end user experience and therefore be undesirable. Therefore it is of most interest to examine the relationships between the error thresholds that garner above acceptable perceptual ratings. Clearly future work is required to examine if smaller thresholds will produce similar lookup tables.

Additionally, previous work suggested that the perceptual rating would fall below acceptable around 8 tgu for a fast paced entity using a spatial metric [8]. In this experiment the motion accuracy falls below acceptable just before 7.5 tgu, which is in keeping with our previous findings. The entity used in this experiment was slightly slower than that of our previous work and as a result slightly lower acceptable spatial threshold is to be expected.

3.2 Experimental Results for Continuous Feedback

This section focuses on the continuous data collected during each video. The results are compared to those presented in Section 3.1. During each video the continuous feedback could be at one of three ratings, *Poor*, *Ok*, or *Good*. The average percentage time per each score was calculated for each scenario and is shown in Table 3.

Scenario	Good %	Ok %	Poor %
Spatial 6	55.72	38.59	5.68
Spatial 7.5	58.08	27.46	14.44
Spatial 9	54.58	34.01	11.4
Spatial 10.5	50.16	26.46	23.36
Time Space 1	70.87	26.69	2.42
Time Space 2	54.86	35.3	9.82
Time Space 3	51.17	24.5	24.31
Time Space 4	52.44	26.08	21.46
Time Space 5	48.69	15.51	35.78
Time Space 7	49.92	15.35	34.72

Table 3: Breakdown of the Average Time Duration for each Continuous Rating Scenario

As expected the amount of time with a *poor* rating increases as the thresholds increase. Interestingly for the spatial metric the amount of time with a *poor* rating never goes above 24%, whereas for a time space threshold anything above and including 3 tgu gives a higher percent for the *poor* rating. Surprisingly the amount of time with a *good* rating remains high regardless of metric or threshold, generally above 50%. Analysing the data further reveals that most of the negative ratings occur around the two corners of the track. A high rating is maintained during the straights and into the early part of each bend.

Despite spending a relatively long period of time with an acceptable rating, a subject may rate the player smoothness and motion accuracy scores as unacceptable. For example, a spatial error of 9 tgu gives a perceptual rating of *poor* for the motion accuracy but the continuous feedback is rated as acceptable 90% of the time. This would indicate that if a large enough error occurs in a simulation, even for a small amount of time, it significantly impacts an end user's experience. This highlights the need for appropriate error thresholds. If an error threshold is chosen on the grounds that it 'appears' suitable, without any research into the end user experience, then it may be acceptable for a large proportion of the time but still be considered a bad system and ultimately be dismissed by potential users.

4. CONCLUDING REMARKS

This paper has shown how psycho-perceptual measures can be used as a tool to garner feedback on various entity update scenarios. Dead reckoning was examined for both spatial and time space metrics under various error thresholds. The resultant perceptual feedback was analysed, which highlighted some interesting issues.

In keeping with our previous work, the perceptual acceptability for a spatial metric falls between 6 and 7.5 tgu for this application. A perceptual lookup table was created that outlined equivalent spatial and time space thresholds. Similarly, the mean squared error for both the spatial and time space metrics were calculated. The two lookup tables were then compared.

Except for the smallest threshold value the two lookup tables did not match. For large error thresholds the finite nature of the perceptual scale results in an upper limit on the potential perceptual rating, whereas the mean squared error can always increase. This results in a perceptual graph that will decrease until it reaches its upper limit, *extremely poor*, whereas the graph for the mean squared error will always increase. Interestingly the smallest threshold value appears to match. This may indicate that, for above acceptable ratings, the lookup tables will match, which would be the area of most interest to developers. Future work will examine smaller error thresholds to determine if this is the case. It is, to a degree, irrelevant to calculate the mean squared error for large error thresholds if it is going to be considered extremely poor by the end user.

Finally the continuous feedback data was analysed. As expected larger error thresholds resulted in more time with a *poor* rating. However, for both metrics, the amount of time with at least an acceptable rating never falls below 64%, yet the resulting player smoothness and motion accuracy

scores were considered unacceptable. It appears large errors, even for a relatively small period of time, result in dissatisfaction for the end user. Error thresholds that are arbitrarily chosen may work most of the time but may ultimately result in a poor end user experience. Also as such thresholds are typically static, there is a need for research into adaptive error thresholds, which future work will focus on.

5. ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland and Enterprise Ireland under grant no. IRCSET/SC/04/CS0289.

6. REFERENCES

- [1] IEEE (1993). "IEEE Standard for distributed interactive simulation – application protocols", in IEEE Std 1278 –1993, I.C. Society, New York, 1993.
- [2] Aggaraval, S., H. Banavar, A. Khandelval, S. Mukherjee and S. Rangajan, "Accuracy in Dead-Reckoning based multiplayer games", SIGCOMM'04 Workshops, Oregon, USA, 2004.
- [3] Delaney, D., T. Ward and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey - Part I." Presence: Teleoperators and Virtual Environments 15(4), August 2006, pp 218-234.
- [4] Delaney, D., T. Ward and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey - Part II." Presence: Teleoperators and Virtual Environments 15(4), August 2006, pp 465-482.
- [5] Delaney, D., T. Ward and S. McLoone, "Reducing update packets in distributed interactive applications using a hybrid model", 16th International Conference on Parallel and Distributed Computing Systems, Reno, USA, 2003, pp 417-422.
- [6] Singhal, S., and M. Zyda, "Networked virtual environments: design and implementation.", ACM Press SIGGRAPH Series, Addison-Wesley, Reading, Massachusetts, 1999.
- [7] 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 Simulations, 14(1), January 2004.
- [8] Marshall, D., S. McLoone, T. Ward and D. Delaney, "Statistical Determination of Hybrid Threshold Parameters for Entity State Update Mechanisms in Distributed Interactive Applications", Forthcoming publication in DS-RT'06, Spain, 2006.
- [9] Kenny, A., S. McLoone, T. Ward and D. Delaney, "Using User Perception to Determine

Suitable Error Thresholds for Dead Reckoning in Distributed Interactive Applications", IEEE Irish Signals and Systems Conference, Dublin, Ireland 2006, pp49-54.

[10] Defanti, T. and R. Fraser, Teleimmersion in the grid: blueprint for a new computing infrastructure, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. pp. 131-155, 1998.

[11] Marshall, D., D. Delaney, S. McLoone and T. Ward, "Exploring the spatial density of strategy models in a realistic distributed interactive application", DS-RT'04, Budapest, Hungary, 2004, pp 210-213.

[12] Miller, G., "The magic number 7 plus or minus 2", Psychological Review, USA, 1956.

7. AUTHOR BIOGRAPHIES



Alan Kenny graduated from NUI Maynooth in computer science and software engineering in 2004. Having spent a short stint working as a web developer he returned to the University to pursue a PhD in the area of psycho-perceptual measures in networked games.



Dr. Séamus McLoone is a member of the distributed interactive applications group (DIAG) at the Department of Electronic Engineering in the National University of Ireland Maynooth. His research interests vary from multiple model approaches to non-linear system identification to reducing and masking the effects of latency in distributed interactive applications.



Dr. Tomas Ward is a member of the distributed interactive applications group at the NUI Maynooth. His main interests are in assistive technology and data management techniques for distributed shared applications. Spare time is spent on pushing out his personal best on Konami DrumMania V.



Declan Delaney received his B.E. and M.Eng.Sc. from University College Dublin in 1991 and 1998 respectively. His PhD was awarded in 2005 by the National University of Ireland, Maynooth. From 1992 to 2000 he worked as a software developer and subsequently as a project manager. At present he works in the area of Knowledge Management for an Italian multinational in Milan.

