

Using User Perception to Determine Suitable Error Thresholds for Dead Reckoning in Distributed Interactive Applications

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 @eeng.nuim.ie*

Entity state update mechanisms are readily employed in Distributed Interactive Applications (DIAs), particularly in networked games. These mechanisms use prediction techniques in order to reduce the number of update packets sent across the network, while maintaining a high level of consistency from the remote user's point of view. These mechanisms only send update packets when the local user's actual behaviour differs from the predictive behaviour by a certain value, often referred to as the error threshold. In practice, this value is arbitrarily chosen and typically reflects what 'appears' to be suitable. It has been illustrated in various other media that psycho-perceptual measures can be used to greatly improve compression techniques, while maintaining satisfactory end-user experience. The best example of this is the MP3 compression used in audio. This paper describes a preliminary study designed to collect information relating to a subject's perception of a networked computer game. The main motivation behind this work is to investigate if psycho-perceptual measures can be used to obtain appropriate error threshold measures for entity state prediction mechanisms. Here, we employ Dead Reckoning as it is the simplest and most commonly used of these mechanisms in distributed gaming. The experiment outlined in this paper attempts to determine if an error threshold can be chosen from the users' perception, where user feedback is determined via linguistic variables. Furthermore, the effects of convergence, the speed of the entity and the shape of the entity trajectory are also examined from a psycho-perceptual viewpoint. The results are presented within.

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

I INTRODUCTION

Interactive online computer games are an important class of Distributed Interactive Applications (DIAs) and one of the fastest growing aspects of the internet [1]. However, these games generate a large quantity of data that must be communicated across the physical network between participating game nodes in order for the game state to be replicated at remote nodes. Unfortunately, there are several fundamental issues that make it difficult to maintain an up-to-date accurate model at remote nodes, including limited network bandwidth and the problems associated with latency and its variance, known as jitter. In the latter

case, latency and in particular jitter mean that there will be a delay between the transmission of data at one node and the play out of that data at the recipient's node, which leads to inconsistencies in the game state at the remote node. Hence, the determination of new techniques to reduce the amount of data that needs to be transmitted in order to maintain a consistent viewpoint for all participants has emerged as a key research area in distributed computer games.

One of the most popular techniques used to date is the entity state update prediction mechanism known as Dead Reckoning. This was introduced as a Distributed Interactive Simulation (DIS) standard [2]

and has been used commercially in games such as Doom 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. Other packet reduction techniques used in DIAs include the hybrid strategy model [3], area of interest management, data compression and dynamic load balancing [4].

In distributed media, such as audio and video, psycho-perceptual phenomena have been recently employed to reduce the amount of data that needs to be communicated, without impacting on the quality of the end-user experience. The most obvious examples of this are the MP3 standard for audio compression and the DivX standard for video compression. However, in DIAs, such as networked games, the exploitation of psycho-perceptual measures has received relatively limited attention. Previous work has instead focused on the effects of latency and jitter on the performance of users in online games [5, 6, 7].

One of the key factors in all entity update prediction methods is when to send the updated information. In Dead Reckoning an update is sent once a certain 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 in respect to the underlying application.

The focus of the research outlined in this paper is to investigate the effect of different error thresholds on the end-users' experience of a simple game. The goal of this work is to determine if the end-user's perception can be used in the determination of a suitable error threshold for distributed games. The desired threshold would minimize the packet data to be transmitted to each node while at the same time maximize the end-user experience. This paper is part of a larger body of work that hypothesises that knowledge of player psycho-perception can be exploited to reduce the amount of data that needs to be transmitted between users in online games [8].

Here, we describe an experiment that collects user perceptual feedback for different gaming situations. The results obtained are presented and analysed. Furthermore, the use of convergence, the pace of the entity and the trajectory of the entity are all examined from the perceptual viewpoint of the end-user.

The remainder of the paper is structured as follows. Section II provides a brief description of the Dead Reckoning technique. A simple convergence routine is also outlined. Section III 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 in section IV. Finally some conclusions and suggestions for future work are given in section V.

II RELATED INFORMATION

For the convenience of the reader, a brief description of the Dead Reckoning entity state prediction technique is now presented. A simple convergence approach is also outlined [9].

a) *Dead Reckoning*

Under DIS, once an entity is created the information pertaining to this entity is transmitted to all participating remote nodes. Each remote node then attempts to predict this entity movement based on its trajectory data. There are many ways in order to interpolate a player's position. The most basic and common of these is to set the new position to the transmitted position and the new velocity to the transmitted velocity, which is known as first order Dead Reckoning. Another common Dead Reckoning technique is to use the transmitted position and velocity along with the transmitted acceleration information.

The local node also keeps a model of itself under Dead Reckoning, which is continually compared to its actual position. Once the actual position differs from its predicted position by a set amount, known as its error threshold, an update is sent to all the remote nodes informing them of its updated trajectories. Once the remote nodes receive this new information they update their models to reflect the latest transmitted data. In this paper, we simply make use of the basic first order Dead Reckoning model. Further information on Dead Reckoning can be found in [2, 10].

b) *Basic Convergence*

One of the key problems with predicting a remote entity's position is that once the updated position has been received, the remote entity's position has to be rapidly corrected. If the remote entity is moved directly into the new position this can result in a disjoint and unnatural behaviour from the user. Convergence is an attempt to naturally blend the incorrect current player trajectory into the updated player trajectory. Here, we employ a basic convergence routine [9].

Once the new trajectory is received two potential future paths are built, one with the updated velocity from the current point and one with the updated velocity from the updated position. A set of future points is then created between these two paths, with an initial weighting favouring the path created with the current position before moving towards the actual path. This intermediate path is then played out with increased velocity.

III EXPERIMENTATION

In this section we outline the experiment used to obtain user perceptual feedback. The experiment was

specifically designed to examine the performance of the basic Dead Reckoning model for varying error threshold values.

a) *Video Clips*

In order to obtain the required feedback, a set of video clips was created. These clips consisted of a pre-recorded movement of a ‘bot’, or a computer controlled entity, as it navigated a very simple gaming race track. The bot’s movement attempts to replicate the ideal movement and is determined by a Dead Reckoning model with a varying set of error threshold values. The ideal movement in this case is where an entity follows the exact shape of the race track in a smooth fashion, i.e. with no latency or jitter. In the case of the bot, both latency and jitter were introduced into the model to simulate a realistic networked game state, as seen by a remote user.

It has been shown that the average transmission times fall within the region of 200ms [11]. In other work, it has also been shown that the end-user’s experience becomes unsatisfactory for a level of latency between 150ms and 200ms. Hence, here we employed a latency of 200ms. Jitter was set as a random element with a maximum limit set to $\pm 10\%$ of the latency value.

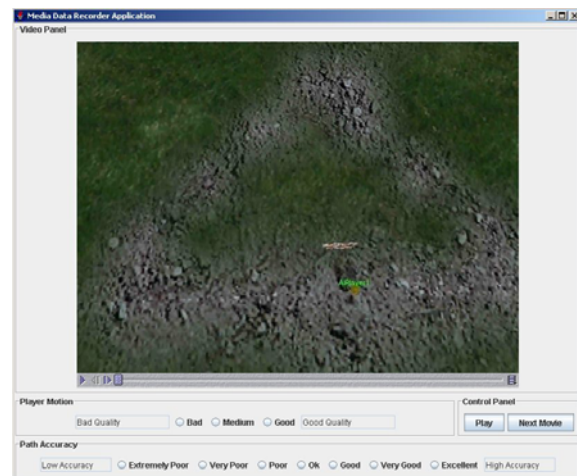
The set of video clips used, employed two different race tracks. These were created in order to investigate the effects, if any, of varying the shape of the track on the user’s experience. The first track was triangular in shape, as can be seen in Figure 1 (a). This was chosen as it contained acute angles, an undesired scenario for Dead Reckoning. The second track was elliptical in shape and reflects a realistic simple course for a racing game. It also highlights natural nonlinear motion. Part of this track is illustrated in Figure 1 (b). A plan view of the tracks was created, as it gave the user total information regarding an entity’s current position.

The pace of the entity, as it traversed a given track, was also varied for different video clips. Two different speeds were used, namely fast or slow. This allowed us to see if the end-user experience was affected by the speed at which entities moved.

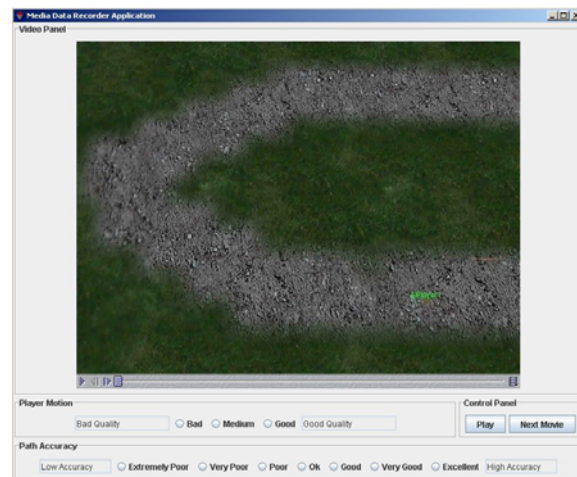
Finally, it was also decided to investigate the usefulness of convergence and to, once again, examine its effect on the end-user experience. Thus, convergence was used to smooth the updates in some of the video clips, but not in others. As detailed in section II, point-to-point convergence was used, as this is the standard in DIAs [2]. In the case of no convergence, the entity was simply instantaneously reset to the updated position.

The Torque Game Engine [12] was used to create the Dead Reckoning model. Torque is an industrial game engine that has been used to produce many games including the award winning Tribes 2. The different error thresholds for the experiment were measured in Torque World Units, which denotes the

spatial difference between the real and the predicted positions. The various games scenes were recorded as AVI files using FRAPS (www.fraps.com), a utility specially designed for recording game footage.



(a)



(b)

Figure 1: Java media data recorder playing (a) a triangular and (b) an elliptical racing track

b) *User Feedback*

The aim of the experiment is to obtain feedback regarding the accuracy of the path followed by the bot. In other words, how realistic is the path travelled by the bot; how similar is it to the actual path? Unfortunately, preliminary testing showed that users’ opinions were significantly influenced by the motion of the bot. For example, although the bot could follow the path almost perfectly, it nevertheless appeared to ‘jump’ from time to time, as a result of poor convergence. This issue is discussed later in the paper. In order to alleviate this problem, the feedback from the subjects was obtained in two parts.

Firstly, they had to indicate the degree to which they felt the model (i.e. the actual bot’s movement)

accurately followed the given path (either triangular or elliptical). Secondly, they had to indicate the degree to which they felt the bot's motion affected their perceptual experience, i.e. how realistic was the motion itself. A Java application, shown in Figure 1, was developed that allowed a subject to watch a movie and then rate it in terms of path accuracy and quality of motion. Here, each subject was asked to keep their judgement until the end of each movie clip. This application could also be used to obtain continuous feedback for longer movies, if necessary.

Linguistic variables were used to obtain subject feedback. The path accuracy variables were *Extremely Poor*, *Very Poor*, *Poor*, *Okay*, *Good*, *Very Good* and *Excellent*. A linguistic scale was used in order to avoid the inherent difficulty of trying to exactly quantify a specific level of accuracy. Furthermore, a seven-point scale was chosen as it has been shown that humans think comfortably in terms of seven [13]. The quality of the bot's movement was selected from the list *Bad*, *Medium* or *Good*. As the experiment was primarily aimed at examining the path accuracy, only a coarse measurement for the quality of the player motion score was examined.

Finally, some general information, particularly in relation to gaming experience, was acquired for each participant.

c) Experimental Setup

At the start of the experiment each subject had the task explained to them and was shown two sample videos for illustration purposes. Once they understood the task, the tests began. This consisted of the subject watching the bot's movement, modelled under perfect conditions, followed by a set of videos showing the model under various conditions. Note, in the perfect condition, the error threshold is set to zero and there is no latency or jitter. The subjects were shown this perfect play out as a reference point to judge the remaining videos.

	Convergence	Track	Pace
<i>Scenario 1</i>	No	Triangle	Slow
<i>Scenario 2</i>	Yes	Triangle	Slow
<i>Scenario 3</i>	No	Triangle	Fast
<i>Scenario 4</i>	Yes	Triangle	Fast
<i>Scenario 5</i>	No	Ellipse	Slow
<i>Scenario 6</i>	Yes	Ellipse	Slow
<i>Scenario 7</i>	No	Ellipse	Fast
<i>Scenario 8</i>	Yes	Ellipse	Fast

Table 1: List of possible video scenarios

The experiment consisted of a total of 72 video clips, 36 for each of the race tracks. For each track, there were 18 fast paced videos and 18 slow paced ones. Each set of 18 clips contained 2 perfect play

outs, 14 unique ones and 2 repeated play outs. The play list was completely random. Each subject watched each video once. At the end of each one, the subject was required to provide feedback, as indicated in the previous subsection. The experiment lasted approximately half an hour. For ease of reference, the full list of possible scenarios is shown in Table 1. Furthermore, for the sake of completeness, the corresponding number of Dead Reckoning update packets is given in Table 2. The results are presented and analysed in the next section.

DR Error	2	4	6	8	10	12	14
<i>Scenario 1</i>	25	7	6	5	4	4	4
<i>Scenario 2</i>	25	9	7	5	4	4	4
<i>Scenario 3</i>	35	35	31	15	11	8	8
<i>Scenario 4</i>	35	35	29	14	11	9	9
<i>Scenario 5</i>	68	11	9	7	7	7	7
<i>Scenario 6</i>	68	11	9	7	7	7	7
<i>Scenario 7</i>	62	61	56	26	14	10	10
<i>Scenario 8</i>	62	54	50	26	17	13	13

Table 2: Number of Dead Reckoning (DR) update packets for different error thresholds

IV RESULTS AND ANALYSIS

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

The average results of all subjects are illustrated in Figures 2 through to 5. Figures 2 and 3 represent the respective plots of the path and player motion accuracy for the triangular track, scenarios 1 to 4. Similarly, plots for the elliptical track, scenarios 5 to 8, are given in Figures 4 and 5.

The results in all the Figures generally show a very rapid decline in satisfaction for slow paced action in contrast to the fast paced movement, where the decline is much less severe. The accuracy rating falls below an acceptable level (in this case, the acceptable level is *Okay*) between 4 and 6 Torque world units for both the triangular and elliptical paths for the slow movement. On the other hand, the accuracy remains acceptable up to 8 and 10 game units respectively for the fast pace scenario. Similarly, the motion quality falls below an acceptable level (in this case, *Medium*) between 4 and 6 units for the slow movement and between 8 and 10 for the fast movement. These results suggest that a higher error threshold can be employed for fast paced racing-style scenarios.

The shape of the path taken does not appear to significantly affect the results. In all cases, the threshold error is more affected by the speed of the entity rather than the path of its movement. However, having said this, it should be noted that, in some instances, a number of the higher error thresholds exhibit greater acceptability than their lower counterparts. For example, in Figure 2, an error threshold value of 12 or 14 appears to perform better for the triangular track than that of 10. Clearly, this is contrary to what we expected. At first glance this appears to be linked to the shape of the course, as results for the elliptical path show no such anomaly. However, the authors feel that this anomaly is associated with poor convergence. Further investigation is required.

The results also show that the model with convergence is less accurate than that without convergence. Again this is contrary to what we expected. In the case of the fast paced elliptical track, the model with convergence performs marginally

better than the model without convergence. This is simply because the shape of the track is more suited to the convergence approach adopted here. In other words, the entity movement appears smoother because there are no acute angles to contend with. This is not the case for the triangular-shaped track.

Figure 6 shows the shape obtained for the triangular path for two Dead Reckoning models, one using convergence, the other using none. In both cases, the error threshold is set to 4 game units. From Figure 2, we see that this particular value indicates a poor convergence performance for a fast pace, although Figure 6 would suggest that this should not be the case. It appears that a sudden jump to the correct position is more favourable to the user than a smooth transition over a longer period of time. Of course, the latter serves to highlight the model inaccuracy, while the former affects the quality of the player motion, although this is not supported by Figure 3. This issue requires further study.

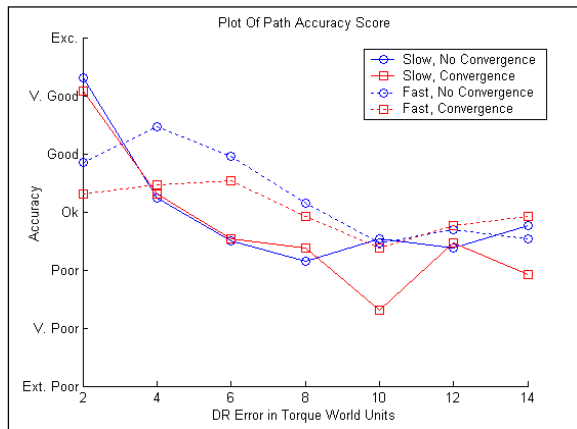


Figure 2: Path Accuracy Rating for Scenarios 1 to 4, Triangular Path

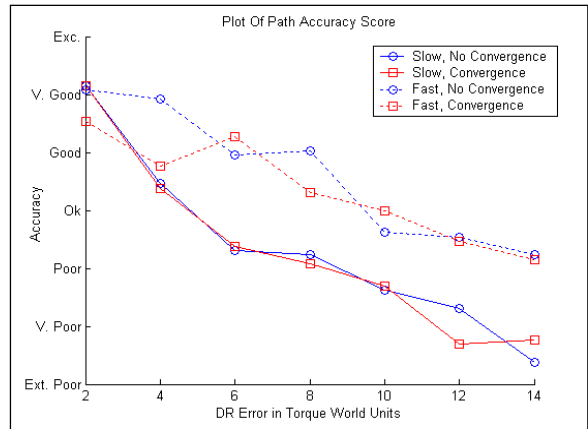


Figure 4: Path Accuracy Rating for Scenarios 5 to 8, Elliptical Path

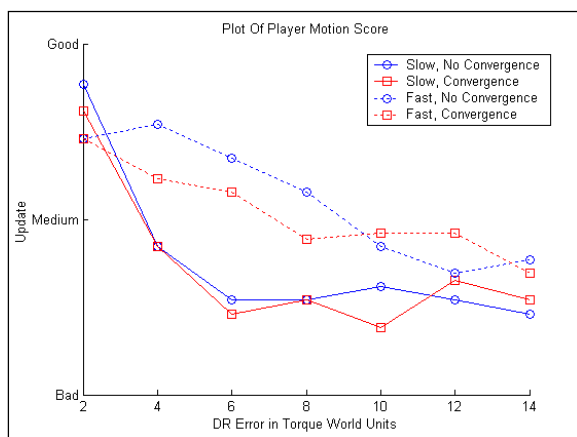


Figure 3: Player Motion Rating for Scenarios 1 to 4, Triangular Path

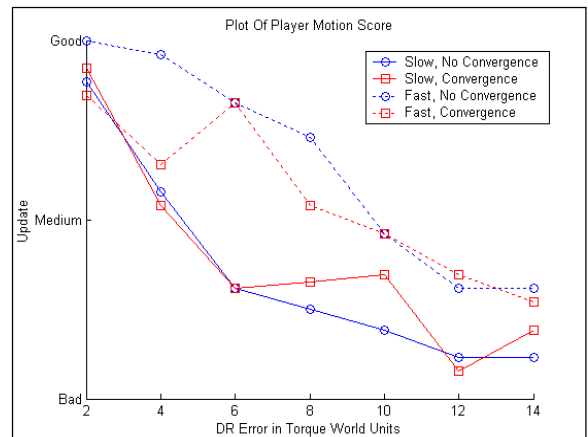


Figure 5: Player Motion Rating for Scenarios 5 to 8, Elliptical Path

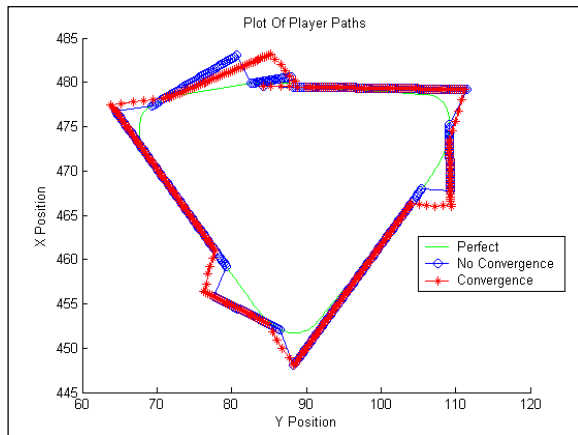


Figure 6: Path trajectory obtained for two Dead Reckoning models, one using a convergence approach, the other using none. The error threshold is 4 Torque world units in both cases.

Another issue with convergence and the triangular path was observed. The point-to-point convergence algorithm struggled with the sharp changes in this path. This is clearly evident from the number of update packets shown in Table 2. In the slow-paced case, the use of convergence resulted in more update packets being sent than for smaller threshold values. This is also reflected by the end-users' experience, as illustrated in Figure 3. A possible solution to this problem of dealing with acute angle position updates is through the use of a pre-reckoning algorithm, as outlined in [14].

V CONCLUDING REMARKS

This paper examined a novel way to determine an appropriate Dead Reckoning error threshold by utilising a user's perception of a gaming situation. In order to achieve this, an experiment was developed that recorded a user's linguistic feedback of various game scenarios. The collated results tentatively suggest that an error threshold for a fast paced game could potentially be almost double that of a slow paced game, while eliciting a similarly acceptable end-user experience. It should also be noted that for fast paced scenarios, in this paper, there is little gain in terms of packet reduction for error threshold values greater than ten Torque world units. Of course, this is application dependent.

The results also show that the issue of convergence is far from solved and that a suitable routine needs to be developed. In fact, it is likely that a set of suitable convergence routines may be required, in order to accommodate the variety in existing games.

Future work will focus on convergence as well as examining different entity prediction techniques, such as the Hybrid Strategy Model [3]. Additionally, an online experiment will be developed to allow for a greater participation. This will also help remove any

outlying data that may occur from having a small data population. Finally, the scenarios detailed in this paper only deal with path motion within a game, which is simply one of many key elements within a networked game. Other elements, such as player interaction, will also be considered from an end-user's perceptual viewpoint.

VI ACKNOWLEDGEMENTS

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

VII REFERENCES

- [1] D. Delaney, A. Kerr and D. Gallagher, "Game for it", *The Engineers Journal* 58(04), pp. 244-249, 2004.
- [2] IEEE (1993). "IEEE Standard for distributed interactive simulation – application protocols", in *IEEE Std 1278–1993, I.C. Society*, New York, 1993.
- [3] D. Delaney, 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.
- [4] S. Singhal and M. Zyda, "Networked virtual environments: design and implementation.", *ACM Press SIGGRAPH Series*, Addison-Wesley, Reading, Massachusetts, 1999.
- [5] L. Pantel and L. Wolf, "On the suitability of Dead Reckoning schemes for games", *NetGames*, Braunschweig, Germany, 2002.
- [6] N. Sheldon, E. Girard, S. Borg, M. Claypool and E. Agu, "The effects of latency on user performance in Warcraft 3", *NetGames*, Redwood City, USA, 2003.
- [7] G. Armitage, "An experimental estimation of latency sensitivity in multiplayer Quake 3", *International Conference on Networks*, Sydney, Australia, 2003.
- [8] H. Koesling, A. Kenny, D. Delaney, S. McLoone and T. Ward, "Decision-making in computer games: the Rubicon model and visual attention", *13th European Conference on Eye Tracking*, Munich, Germany, 2005.
- [9] D. Delaney, S. McLoone and T. Ward, "A novel convergence algorithm for the hybrid strategy model packet reduction technique", *IEE Irish Signals and Systems Conference*, Dublin, Ireland, 1-2 September, pp.118-123, 2005.
- [10] S. Aggaraval, H. Banavar, A. Khandelval, S. Mukherjee and S. Rangajan, "Accuracy in Dead-Reckoning based multiplayer games", *SIGCOMM'04 Workshops*, Oregon, USA, 2004.
- [11] T. Defanti 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.
- [12] D. Marshall, 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.
- [13] G. Miller, "The magic number 7 plus or minus 2", *Psychological Review*, USA, 1956.
- [14] T. Duncan and D. Gracanin, "Pre-reckoning algorithm for distributed virtual environments", *Proc. of the Winter Simulation Conference*, 1087- 1093, 2003.