Using Cursor Prediction to Smooth Telepointer Jitter

Carl Gutwin, Jeff Dyck, and Jennifer Burkitt
Computer Science Department, University of Saskatchewan
Saskatoon, Canada, S7N 5A9
+1 306 966-8646
carl.gutwin, jeff.dyck, jen.burkitt @usask.ca

ABSTRACT

Telepointers are an important type of embodiment in real-time distributed groupware. Telepointers can increase the presence of remote participants and can provide considerable awareness information about people's locations and activities. However, the motion of a telepointer is often disrupted by network jitter. Although some strategies exist for dealing with jitter, none of these techniques are able to restore the immediacy and smoothness of a real cursor. In this paper we investigate the use of prediction – commonly used in networked simulations and games – to reduce the effects of jitter on telepointer motion. To determine whether prediction can be effective for improving telepointers, we carried out two experiments that tested the effects of different prediction schemes (some real and some artificial) on people's ability to interpret telepointer gestures. These studies show that although cursor prediction is still a difficult problem, there are both potential performance improvements, and definite preference advantages. Our studies suggest that telepointer prediction should be routinely used to increase the immediacy and naturalness of remote interaction, and suggest that prediction can also improve interpretation in certain situations.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Tools and Techniques—User interfaces; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—Computer-supported cooperative work.

General Terms

Performance, Design, Experimentation, Human Factors.

Keywords

Groupware, real-time groupware, telepointers, dead-reckoning, prediction, network delay, jitter.

1. INTRODUCTION

Telepointers are a common means for representing participants in real-time distributed groupware. Embodiments such as telepointers are extremely valuable in that they can show presence, location, and activity, and can also enable gestural communication (e.g. [4,9]). For example, people can watch the movement of another person's telepointer to determine what they are doing, or can see explicit gestures produced by the other person.

For telepointers to convey a sense of natural presence, or to adequately show activity and gesture, the motion of the telepointer must be smooth and consistent [12]. The telepointer position must be updated at a high enough frequency for people to be able to adequately interpret the cursor's motion, and this update frequency must remain consistent to ensure that the pacing of the movement closely matches the original motion of the remote user's cursor.

However, these requirements are rarely met by real-world wide area networks such as the Internet. In particular, when there is *jitter* in the network – that is, variance in the end-to-end latency between two groupware applications – telepointers move in a halting and jumpy fashion. This type of movement is immediately noticeable, and (depending on the amount of jitter) can cause substantial problems for interpreting the telepointer's movement.

Although methods have been proposed for reducing jitter by buffering [12] or compensating for jitter with visual traces [10], these techniques are limited. In particular, they cannot maintain both the immediacy and the naturalness of the original cursor motion, and groupware users must adapt their interaction as a result. In this paper we explore a different approach that attempts to achieve smooth and natural telepointer motion without increasing latency. This approach uses telepointer prediction: whenever a telepointer position is unavailable due to network jitter, the receiving system will calculate a new position, thus (artificially) maintaining the pacing of the telepointer's motion.

In this paper we consider the issue of whether prediction in general is an effective approach (rather than which prediction technique to use). To test the effectiveness of telepointer prediction, we carried out two experiments. The first compared a basic dead reckoning algorithm to a 'normal' jittered telepointer. Participants were asked to identify simple gestures that were drawn by either the predicted or the unpredicted telepointer. This experiment showed that although participants preferred the predicted telepointer, prediction did not improve interpretation, because at levels of jitter where gesture interpretation begins to become difficult (above about 200ms), prediction error was also high.

These results raise the question of whether any prediction scheme could succeed in improving gesture interpretation. Prediction error appeared to be the main factor, which leads us to our second study. This experiment examined the question of exactly how accurate a prediction system would have to be in order to show a performance improvement. We used a Wizard-of-Oz method to manipulate the error of an artificial prediction technique. Using simple linear prediction as the baseline, we produced three artificial predictors with controlled error rates. We once again asked participants to interpret telepointer gestures under several different prediction and jitter conditions. Our results show that when the jitter period is 320ms, a prediction scheme would have to have less than 40 pixels mean maximum error – about 25% less error than linear prediction – to significantly improve gesture interpretation over no prediction.

These empirical results are the first lower bound we are aware of that designers can use when considering whether a specific prediction technique will be effective in reducing the interpretation problems that are caused by network jitter.

In addition, the idea of telepointer prediction is one that presents a number of additional possibilities for improving networked groupware. In particular, we argue that the display of a telepointer can be de-coupled from the sampling and sending of the position information. Decoupling could lead to several benefits. For example, it may be feasible to considerably reduce the number of telepointer messages sent between groupware systems without unduly affecting the receiver's perception of the telepointer – since even basic prediction algorithms work quite well for short periods of time.

In the following sections we provide more detail on the use of telepointers for gesture in groupware, the problem of jitter, and the idea of motion prediction. We then report on the two experiments and discuss the implications for the design of telepointers in real-time groupware.

2. GESTURES IN GROUPWARE

Gestures in groupware are frequent and are used for a variety of purposes. In general, information may be communicated through gestures in two ways, either explicitly or implicitly.

Explicit gestures are intentional actions designed to convey a particular message to another person. There are several types of explicit gesture: for example, pointing to indicate objects, areas, and directions [19], drawing to show paths, shapes, or abstract figures [3], describing to show orientations, distances, or sizes [3], or demonstrating to act out the use or operation of an artifact [18]. There are other more specialized types as well, such as the emblem, where a gesture stands for a particular word or phrase (e.g. a checkmark for "OK").

The second type of gesture is that which communicates implicitly and unintentionally – where others pick up information simply from watching another person's movements and actions. Many kinds of activity have characteristic and recognizable motions that, although not intended to convey information, can easily be seen and interpreted by another person. For example, the back-and-forth motion of erasing a drawing with a pencil eraser can be understood even from a distance. These implicit gestures are also important for the smooth operation and coordination of a group, since they provide valuable awareness information about others' actions and activities [17].

In the studies described below, we focus on explicit gestures – in particular, drawings and demonstrations that involve tracing a shape on the work surface.

3. TELEPOINTER JITTER

There are two main types of delay in groupware systems: latency and jitter. Latency is the lag time between the occurrence of an event on a local machine (e.g. movement of the mouse) and display of that event on a remote machine (e.g. movement of the telepointer). Latency has been shown to cause problems in a variety of group interactions (e.g. [16,6]). However, simple latency is not a major problem for the interpretation of gestures, as long as all messages are delayed by the same amount.

The second type of delay – jitter – is much more problematic for gesture interpretation. Jitter is variance in latency, causing some messages to arrive too far apart, others too close together (see Figure 1). Jitter is only an issue for streams of information that have a meaningful spacing to begin with, such as voice data or cursor movement. Jitter occurs because network traffic changes from moment to moment, causing variable slowdowns, loss, and bottlenecks. To a viewer, the characteristic effect of jitter is halting, jerky motion. This effect can be broken down as follows:

1. The sender produces a regular stream of messages (e.g. pointer positions every 20 ms);
2. Something in the network (e.g. an unrelated traffic surge) causes a delay in the stream of position messages;
3. As a result of the delay, several messages 'pile up' and arrive at the same time at the receiver;
4. Processing this group of messages results in several coincident requests to draw the telepointer (assuming that no application-level buffering is done);
5. The display system draws each position, but the screen is refreshed with the next position too quickly for motion to be discerned by the user.
6. The viewer sees the telepointer jump from its initial position, where it had been frozen since the delay began, to the current location, without seeing the intermediate positions.

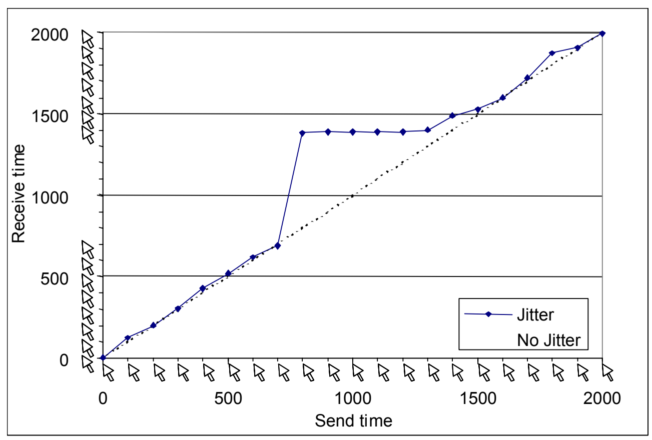

Figure 1. Example of pileup caused by network bottleneck in a stream of UDP messages sent at 100ms intervals. Messages 1-8 are all basically on time and are spaced appropriately. Messages 9-14 are all delayed by a network bottleneck, and arrive at the receiver at the same time. The line of cursors at bottom shows the original spacing of pointer positions; at left shows which positions are drawn.

Jitter has primarily been studied in streaming media, where is has been shown that people notice even very small variations in playback of sound and video files. Jitter has also been shown to cause problems for certain types of group interaction in groupware. For example, jitter delays of more than about 300ms make it more difficult for people to predict where another person's telepointer is moving [12]; delays of more than about 200ms make gesture interpretation more difficult [11].

4. MOTION PREDICTION

Techniques for predicting an object's location based on past positions has been explored in two main ways: first, as a way to maintain consistency in distributed simulations and networked games, and second, as a way to improve targeting in single-user interfaces. Both of these efforts are valuable in considering telepointer prediction, since the former is concerned with overcoming network delays, and the latter is specifically concerned with cursors.

Motion prediction in distributed simulations and games is used to maintain a consistent world-view for each participant (e.g. [7,13]. Prediction is needed in these situations primarily because of network delay (both latency and jitter). For example, in a multi-player game, a remote player's avatar may appear to be in one position, but because of network delay, will actually be in a different location (e.g. [14]). When objects or avatars interact (e.g. shoot each other), these inconsistencies can lead to confusion (e.g. clear hits have no effect on the target). Therefore, most real-time networked games use physically-based prediction schemes (generally called *dead reckoning*) to calculate location and motion.

Dead-reckoning has been extremely successful in improving a game-player's interaction with distributed objects (e.g. [5,8]), and this success suggests that the technique may also be effective for telepointers in groupware. However, one caveat is that dead-reckoning is best applied to objects with force-based movement models and strong inertial properties (e.g. ships, vehicles, bodies), since inertia makes motion more predictable. Telepointers are moved through absolute positioning of a mouse, rather than by a force model, and therefore are able to move in more unpredictable ways than the objects typically found in games and distributed simulations.

The second background area, target prediction, is not motivated by the problems of network delay, but does deal specifically with the issues involved in predicting cursor motion (e.g. [2,15,21]). Researchers in this area are interested in improvements to single-user interaction with graphical interfaces; and more specifically, have investigated whether the eventual rest position of a cursor movement can be determined. The reason for making this prediction is to improve targeting time: for example, the user could save time by not having to move all the way to the target, or could have the predicted target expand as the cursor moved toward it.

Some of these systems have been successful in laboratory experiments, and have used several prediction algorithms such as Kalman filters, neural networks, or non-linear regression (e.g. [2]). These results similarly suggest that telepointer prediction may have potential. Again, however, the techniques cannot be simply transferred to the groupware setting. Target prediction is concerned with eventual rest position, rather than path to get there, and is much more accurate with ballistic movements of the pointing device (which again introduce the benefits of inertia). Gestures involve slower controlled motion with more twists and turns that are more difficult to predict, and so it is not clear whether either dead-reckoning or target prediction methods will be effective in smoothing telepointer jitter.

To determine whether telepointer prediction can be effective, we carried out two experiments. The first tested the effects of a real prediction algorithm on people's abilities to interpret telepointer gestures, and the second considered the issue of how accurate any prediction scheme would have to be in order to cause a performance improvement.

5. FIRST STUDY: THE EFFECTIVENESS OF A BASIC PREDICTION TECHNIQUE

Our first study was intended as a pilot to get an initial idea of how a basic dead-reckoning technique would compare against an unaltered (i.e. affected by jitter) telepointer.

5.1 Methodology

5.1.1 Participants and Apparatus

Eight paid participants (4 male, 4 female) were recruited from a local university. All participants were right handed and were frequent users of mouse and windows systems (at least 12 hours/week).

The experiment was conducted on a Dell Inspirion 4100 PC running a custom-built TCL/Tk application, using a 14-inch monitor set to 1400x1050 resolution. The study system presented a series of trials showing gestures drawn by a simulated telepointer.

5.1.5 Gesture types and tasks

The gestures in the study all involved the tracing of a shape or symbol, one common type of gestural communication. To test the interpretability of these shape gestures, we prerecorded a set of 33 shapes (see Figure 2) that included letters, numbers, and simple strokes. The shapes were all approximately 10 cm in height when drawn on the screen. The participant's task was to determine which of the 33 gestures was being shown on the screen; participants chose their answers from an answer sheet that showed all the gestures (similar to Figure 2) and entered their answer on the system.

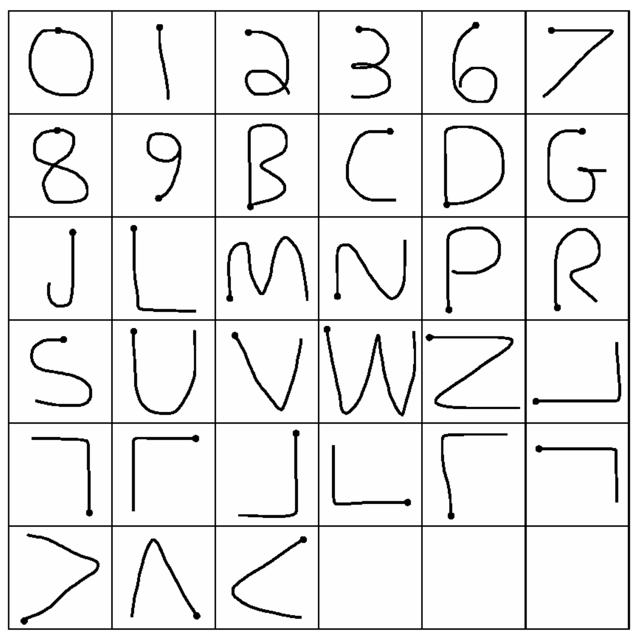

Figure 2. Shape gestures used in the study. The dot indicates where the gesture begins.

5.1.3 Gesture Replay Mechanism

The study used pre-recorded traces of gestures, in order to give each participant a consistent viewing experience. These traces contained time and position information that allowed the replay of the gesture in the experimental system. In addition, we were able to manipulate the replay of the gesture in order to create jitter and prediction conditions.

The replay mechanism used the following rules:

- The position information in the trace are extracted at a fixed rate (as if the remote cursor in a real groupware system was being sampled and sent at a fixed rate – e.g. every 30ms).
- Jitter events are simulated by 'holding back' messages from the trace for a set amount of time (the jitter period), and then processing this group all at once.
- In prediction conditions, messages in this jitter group were ignored during the jitter period, and position information is calculated based on the last known locations.
- At the end of the jitter period, the actual position information is fed into the prediction algorithm, so that the next set of predictions always begin with true position data.
- As soon as one jitter period was completed and the previous position information was delivered, another jitter event would start.
- No correction was carried out; that is, no attempt was made to smoothly connect an erroneous predicted path with newly-received actual position information.

5.1.4 Prediction algorithm

The algorithm used in the study is a basic dead-reckoning scheme similar to that used in distributed simulations. The algorithm predicts the next point based on three values: the most recent absolute position, the most recent change in x and y values (i.e. the velocity), and a rolling average of the recent changes in velocity (i.e. the acceleration).

$$x_{next} = x_{previous} + velocity_x + averageAcceleration_x$$

$$y_{next} = y_{previous} + velocity_y + averageAcceleration_y$$

This technique allows curves to be predicted, and works fairly well at low jitter periods. Figure 3 shows that the scheme is reasonably accurate when predicting motion for 160ms; however, when predicting for 320ms, the errors become considerably larger (Figure 4).

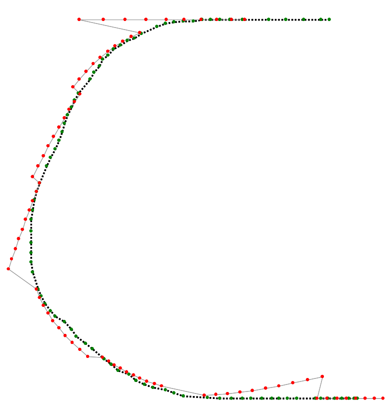

Figure 3. Prediction for shape #10 (see Figure 2), with jitter periods of 160ms. Dashed line and green dots indicate actual shape; grey line and red dots indicate predicted path.

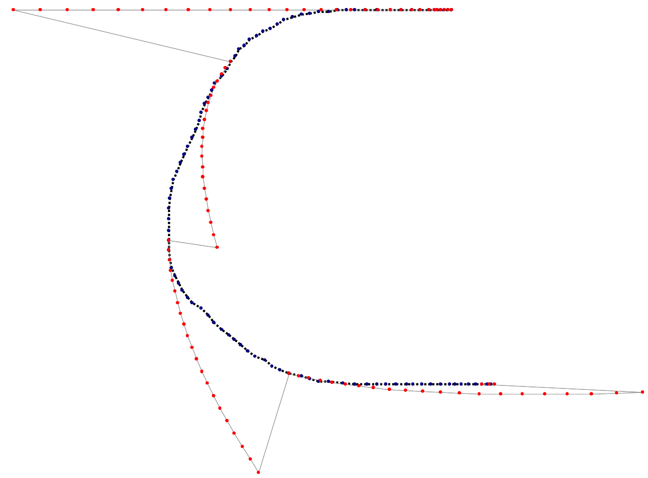

Figure 4. Prediction for shape 10 with jitter period of 320ms.

5.1.5 Procedure

Participants were introduced to the idea of telepointers and network delay, and were shown the experimental system. Participants then completed 60 trials which were grouped into eight blocks. For the lowest and highest jitter periods (considered to be boundary conditions), participants completed five trials per block; for the two middle jitter periods, participants completed ten trials per block.

In each trial, the participant viewed a telepointer movement, and then entered into the system the identification number of the shape that they believed best fit what they had seen. Participants were supplied with a list of the 33 shapes and their corresponding identification numbers. After each answer, the participant could continue to the next trial by pressing the space key on the keyboard. Upon completion of the 60 trials, participants completed a brief post-test. In this test, participants were shown two example gestures from each study condition, and were asked to indicate which example they preferred for telepointer motion.

5.1.5 Experimental Design

The first study compared participants' interpretation accuracy when viewing either a jittered or a predicted stream of telepointer messages. We used a 2x4 within-participants mixed factorial design. The factors were:

- Prediction Type: None (Jitter) or Dead Reckoning (DR)
- Jitter period: 80ms, 160ms, 240ms, 320ms

In total, 480 gestures were shown in the experiment. Data collection included decisions about what each gesture showed (collected by the experiment system), and the post-experiment questionnaire which recorded the participant's preferences.

5.2 First Study Results

Our goal in the first study was to determine whether participants would be more accurate in interpreting gestures when prediction was used, and whether participants would prefer prediction over the corresponding jitter conditions. The results reported below are organized around these two issues.

5.2.1 Accuracy

Figure 5 shows the mean percentage of correct answers for each jitter period, for both prediction and no prediction. ANOVA showed no significant accuracy differences related to the use of prediction ($F_{1,7}=2.24$, $p=0.14$). In addition, as can be seen from the figure, accuracy is lower at jitter periods of 240ms and 320ms.

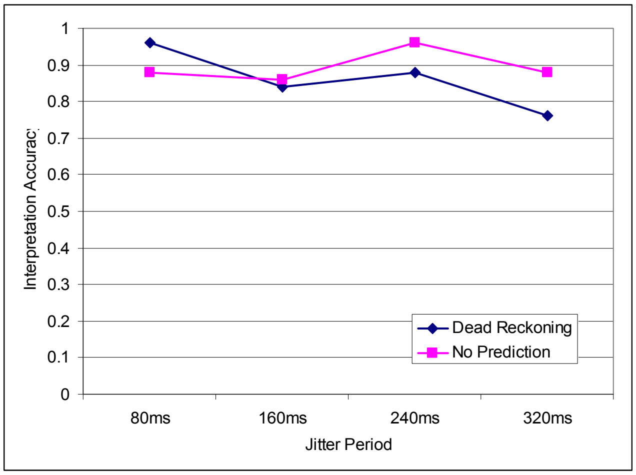

Figure 5. Mean interpretation accuracy for all jitter periods.

It was clear that at lower jitter periods, people do not have much difficulty 'filling in the gaps' caused by jitter. Therefore, in the situations where the dead-reckoning algorithm was accurate, there is very little room to improve people's performance. As the jitter period increases, making interpretation more difficult, the error of the dead-reckoning algorithm also increased, reducing the effectiveness of the technique.

5.2.2 Preference

Table 1 shows participants' preferences for either prediction or no prediction, for each jitter period. At jitter periods of 80ms and 160ms, all participants preferred the predicted telepointer; at 240ms, preference was mixed; and at 320ms, prediction was no longer seen as an improvement over the normal jittered telepointer.

Table 1. Preference results (number of participants) for each jitter period.

Prediction type	80ms	160ms	240ms	320ms
Dead reckoning	8	8	4	0
No prediction	0	0	4	8

6. SECOND STUDY: HOW ACCURATE DOES PREDICTION NEED TO BE?

The first study suggests that it will be difficult for a dead-reckoning algorithm to improve interpretation performance if the error of the predictions grows more quickly than the problems caused by the jitter itself. To determine just how accurate any prediction scheme would have to be to significantly improve performance, we carried out a second study with jitter periods that were larger than those used earlier.

6.1 Methodology

6.1.1 Participants and Apparatus

Eighteen paid participants (13 male, 5 female) were recruited from a local university. All participants were right-handed and were frequent users of mouse-and-windows systems (at least 12 hours/week). Ten of the eighteen participants had some experience playing multiplayer games.

The experimental setup was similar to that used in the first study (Section 5.1.1), except that the software was modified to use a different set of jitter conditions and a different prediction algorithm.

6.1.3 Artificial Prediction Schemes

The prediction algorithm was developed using a Wizard-of-Oz strategy to allow manipulation of the degree of prediction accuracy that the algorithm could attain. Because the system has knowledge of the real future positions of the telepointer (since we have the entire trace), we can determine a set of artificial prediction schemes with varying levels of accuracy.

In the second study, we created artificial predictors with a range of accuracies between simple linear prediction and perfect prediction. Linear (i.e. straight-line) prediction was used as the baseline because it can easily be implemented in any groupware system. With linear prediction, and with foreknowledge of the actual positions, we can manipulate the accuracy of the artificial prediction schemes by translating each predicted point a certain percentage of the distance between the linear predicted point and the actual point (see Figure 6).

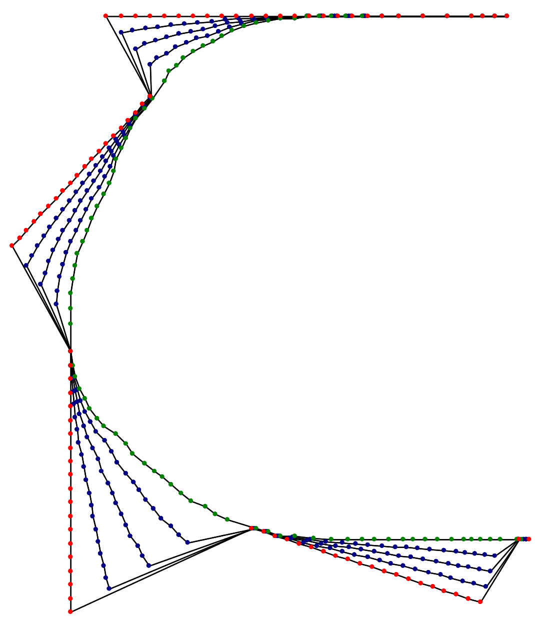

Figure 6. Artificial prediction techniques. Inside line (P0) is the actual shape, outside line is linear prediction (P100). The three lines in between (P75, P50, P25) represent different error amounts compared to linear prediction.

The relative-to-linear categorization that was used to create these prediction schemes, however, is inappropriate for comparison with other techniques. Therefore, we also calculated a measure of their absolute error for each jitter period. We chose *mean maximum error* (MME) as a reasonable indication of absolute error. Mean maximum error is the average distance from the actual position of the most erroneous predicted point in each jitter period. We believe that this measure is more appropriate than overall average error, because it appeared to be the large deviations that caused problems for interpretation in the first study. The MMEs for the five prediction schemes are shown in Table 2.

For comparison, the dead-reckoning algorithm used in the first study has an MME of 188 pixels at 320ms, and of 406 at 480ms.

Table 2. Mean maximum error amounts (in pixels) for prediction schemes used in second study

Scheme	Description	MME (320ms)	MME (480ms)
P100	Linear	160	260
P75	Artificial predictor with 75% of linear error	120	195
P50	Artificial predictor with 50% of linear error	80	130
P25	Artificial predictor with 25% of linear error	40	65
P0	Perfect prediction	0	0

6.1.5 Procedure
Participants were introduced to the study in a similar way to that described above (Section 5.1.5). Participants then carried out a set of 25 practice trials, and then completed 120 test trials grouped into 12 blocks (10 trials in each study condition). Again, each trial involved viewing a remote cursor movement and then identifying the shape that was drawn. Participants were allowed to rest between blocks.

Upon completion of all trials participants completed a brief post-test that was similar to that used in the first study. In the test, participants were shown example gestures from the different study conditions and were asked to indicate their preferences.

6.1.4 Experimental Design
The second study also compared participants' interpretation performance when viewing telepointer gestures with a range of prediction schemes. We used a 6x2 within-participants mixed factorial design. The factors were:

- Prediction Type: None (Jitter), P100 (Linear), P75, P50, P25, or P0 (Perfect)
- Jitter period: 320ms or 480ms

With 18 participants and 10 trials per condition, 2160 gestures were shown in total. Data collection included decisions about what each gesture showed and elapsed time to make each decision (collected by the experiment system), and the post-experiment questionnaire which recorded the participant's preferences.

6.2 Second Study Results
Results from the second study are presented below, organized by interpretation accuracy and response time.

6.2.1 Interpretation Accuracy
Figure 7 shows the mean rate of correct answers for each prediction type. As expected, ANOVA shows a significant main effect of prediction type ($F_{1,17}=19.0$, $p<0.001$). To determine the minimum level of accuracy needed to significantly improve performance over no prediction, we carried out post-hoc t-tests comparing the no-prediction condition to each other scheme. Results of these tests were similar for both jitter periods: there was no difference between no-prediction and P100 (linear prediction), but all of the other schemes showed significant differences (see Table 3).

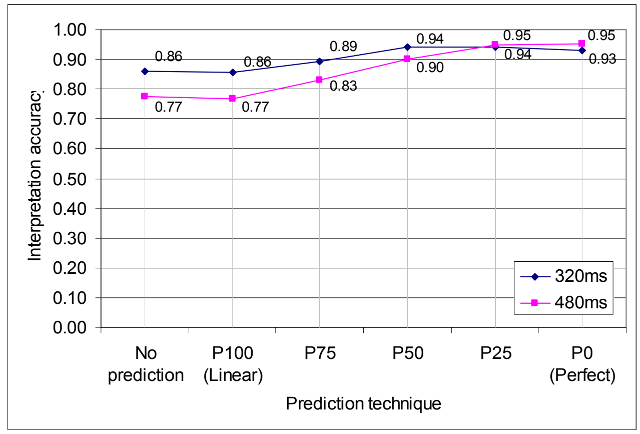

Figure 7. Mean interpretation accuracy for all prediction techniques.

Table 3. T-test comparisons (two-tailed) between no prediction and other schemes, for 320ms data.

Comparison to:	p	Actual difference in interpretation accuracy
P100	0.28	0%
P75	<0.001	3%
P50	<0.001	8%
P25	<0.001	9%
P0	<0.001	9%

The artificial predictor P75 (which had 75% of the error of linear prediction) is the first predictor that is significantly different from no prediction. Using the absolute error values given above (Section 6.1.3), any prediction algorithm could therefore significantly improve interpretation if it had an MME of 40 pixels at 320ms jitter, or 65 pixels at 480ms jitter.

6.2.2 Response Time
We also considered how response times (the time it took participants to enter their answer) varied across prediction types. Figure 8 shows mean response times for each type; the results are similar to those for interpretation accuracy. There was a main effect of prediction type ($F_{1,17}=27.9$, $p<0.001$), and post-hoc t-tests showed that the first predictor to be significantly different from no prediction was P75 ($p<0.05$).

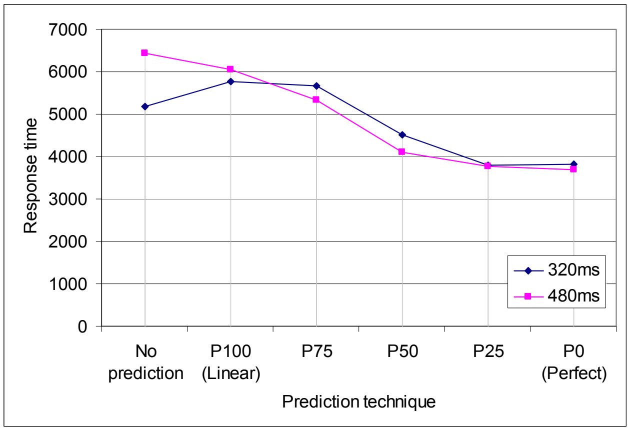

Figure 8. Mean time to answer for all prediction techniques.

7. DISCUSSION

In the following paragraphs, we discuss several issues that arise from the two studies reported above. We first summarize the results, and then consider whether telepointer prediction should be part of groupware systems, either on the grounds of assisting interpretation or on the grounds of user satisfaction. We then examine other potential benefits of prediction, such as the idea of using prediction to make groupware communication more efficient.

The main results from the first study were that basic dead reckoning does not improve gesture interpretation, primarily because prediction error increases along with the jitter period; however, participants all preferred the predicted presentation at low jitter periods. The second study showed that a prediction algorithm would require a mean maximum error of 40 pixels in order to make a significant improvement in interpretation. This error rate corresponds to 75% of the error of linear prediction.

Effectiveness of telepointer prediction. For improving interpretation of gestures, telepointer prediction remains a difficult task. Even though a prediction scheme needs only outperform linear prediction by a small margin, this may be difficult to do. For example, the dead reckoning algorithm used in the first study (a technique that has been successful in many networked games) actually has a higher MME than linear prediction at both 320ms and 480ms. This suggests that telepointers may not be amenable to higher-order models that use velocity and acceleration, and consequently that more complicated prediction schemes (e.g. Kalman filters or non-linear regression) may also have difficulty with telepointer motion during gestures. However, we leave to future work the classification of other prediction schemes in terms of our framework (to facilitate testing, the shape gesture data used in our studies can be obtained from hci.usask.ca/projects/prediction.xml).

However, the studies also show several positive results for prediction. First, the gestures tested in our studies are likely one of the most difficult prediction tasks available (and were chosen for this reason) due to the large number of curves and corners in the shapes. There are several types of telepointer motion that should be easier to predict. For example, moving to a particular point in the workspace, which is also negatively affected by jitter [12], involves much more straight-line motion than the gestures studied here.

Second, the second study showed that the performance of a very simple prediction technique (linear prediction) was surprisingly good at higher jitter periods – in that it was no worse than no prediction. This at least means that prediction need not reduce interpretation performance. This is important, because one of the main benefits of prediction appears to be in increasing user satisfaction. There was a strong preference for the smoothness provided by the prediction techniques (as long as the prediction's errors did not cause more problems than the original jitter). This implies that predicting may be well worth using in groupware, even if it is unclear whether it will improve interpretation.

Other uses of prediction. The success of prediction techniques at low jitter periods raises the possibility of using these mechanisms for a new purpose – not to combat existing jitter, but rather to tune the network distribution and presentation performance of the groupware system. From our empirical results, it appears that a groupware system does not need to send every telepointer position as a separate message: it should be possible to send fewer messages (although still including all data points) and simply predict those points that have been delayed (this essentially adds controlled artificial jitter at the application level). This would allow systems to reduce telepointer messages by a considerable amount (e.g. 5-10 times, based on the data of our first study) – or alternatively, to send a much higher-granularity message stream without requiring additional message traffic.

This idea also suggests a more general approach to the problem of telepointer performance. It is possible that the sending of a telepointer's position can be decoupled from its screen representation, and these two entities can be modeled independently. Using prediction allows groupware to stop treating telepointers as an event-driven system, and avoid some of the problems that occur when events with temporal requirements are sent over inconsistent networks like the Internet. This approach suggests, for example, that each client can sample and send cursor information at its own optimal rate (depending on factors such as processor load); but receiving clients can update the telepointers based on their local redisplay frequency. This could lead to a groupware user experience that is more natural than what has currently been possible.

8. FUTURE WORK

In future work, we plan to continue this research in three ways. First, we will implement and categorize other existing prediction techniques to test their accuracy on our data sets; if any of these have a lower MME than linear prediction, we will test its efficacy in a further user study. Second, we plan to develop the idea of decoupling telepointer display from sampling and distribution; one goal is to build an adaptive system that can optimize the distribution rate and the displayed update frequency based on current network conditions. Third, we plan to consider how prediction techniques work with other telepointer enhancements: for example, with historical traces [10] or delay indicators [4].

9. CONCLUSION

Telepointers are important parts of many groupware systems, both for providing a sense of presence for remote participants, and for conveying awareness information and gestural communication.

However, the quality-of-service requirements for delivering smooth and natural telepointer motion are not often met on networks such as the Internet. In this paper, we proposed the idea of telepointer prediction – based on techniques seen in distributed simulations and games – as a way to smooth telepointers in jittery networks. We ran two studies to explore the effectiveness of telepointer prediction on people's abilities to interpret gestures.

Although the studies show that it may be difficult to successfully improve interpretation, it was clear that people liked the smoother motion provided by prediction techniques. Given that there are simple techniques that (at least) do not appear to degrade interpretation, we suggest that telepointer prediction should be considered as a way to improve user satisfaction with real-time groupware.

9. REFERENCES

[1] Baldwin, J., A. Basu and H. Zhang, Predictive Windows for Delay Compensation in Telepresence Applications, *Proc. 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16-20, 1998.

[2] Baldwin, J, Basu, A., and Zhang, H., Cursor trajectory prediction with a Kalman filter, in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, 1998, pp 2884-2889.

[3] Bekker, M., Olson, J., and Olson, G. Analysis of Gestures in Face-to-Face Design Teams Provides Guidance for How to Use Groupware in Design. *Proc. DIS'95 Symposium on Designing Interactive Systems*, 1995, 157-166.

[4] Benford, S., Bowers, J., Lennart, E.F., Greenhalgh, C., Snowdon, D. User embodiment in collaborative virtual environments. *Proc. ACM CHI'95*, 1995, 242 – 249.

[5] Capin, T., Pandzic, I., Thalmann, D., Magnenat-Thalmann, N., A Dead-Reckoning Algorithm for Virtual Human Figures, *Proc.VRAIS'97* (IEEE Press), Albuquerque, USA, pp.161-168, 1997

[6] Conner, D.B. and Holden, L.S., Providing a Low-Latency User Experience in a High-Latency Application. *Proceedings of 1997 Symposium on Interactive 3D Graphics,* Providence, Rhode Island, April 27-30, 1997.

[7] Diot, C., and Gautier, L. A Distributed Architecture for Multiplayer Interactive Applications on the Internet, *IEEE Networks*, Vol. 13, No. 4, pp.6-15, July-August 1999.

[8] Durbach, C. and Fourneau, J-M, Performance Evaluation of a Dead Reckoning Mechanism, *Proceedings of the Second International Workshop on Distributed Interactive Simulation and Real-Time Applications*, IEEE Press, 1998, Montreal, Canada, pp. 23-32.

[9] Greenberg, S., Gutwin, C., and Roseman, M. (1996), Semantic Telepointers for Groupware. *Proc. OzCHI '96*, Hamilton, NZ.

[10] Gutwin, C. Traces: Visualizing the Immediate Past to Support Group Interaction. *Proceedings of Graphics Interface 2002*, Calgary, 2002

[11] Gutwin, C., and Penner, R. (2002) Improving Interpretation of Remote Gestures with Telepointer Traces. *Proceedings of ACM CSCW 2002*.

[12] Gutwin, C. The Effects of Network Delays on Group Work in Real-Time Groupware. *Proc. 7th European Conference on CSCW*, Bonn, 2001, 299-318.

[13] Macedonia, M., Brutzman, D., Zyda, M., Pratt, D., Barham, P., Falby, J., and Locke, J., NPSNET: A Multi-Player 3D Virtual Environment Over the Internet, *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, 9, Monterey, California, 1995.

[14] Mauve, M., How to Keep a Dead Man from Shooting, *Proc. 7th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services* (IDMS) 2000.

[15] Murata, A., Improvement of Pointing Time by Predicting Targets in Pointing with a PC Mouse, *International Journal of Human-Computer Interaction*, 1998, 10, 1, pp 23-32.

[16] Park, K. and Kenyon, R. (1999), Effects of Network Characteristics on Human Performance in the Collaborative Virtual Environment. *Proc. IEEE Virtual Reality '99*, March 14-17, 1999.

[17] Segal, L. Designing Team Workstations: The Choreography of Teamwork, in *Local Applications of the Ecological Approach to Human-Machine Systems*, P. Hancock, J. Flach, J. Caird and K. Vicente ed., Erlbaum, Hillsdale NJ, 1995, 392-415.

[18] Tang, J. Findings from Observational Studies of Collaborative Work, *International Journal of Man-Machine Studies*, 34(2), 1991, 143-160.

[19] Tatar, D., G. Foster, and D. Bobrow. Design for Conversation: Lessons from Cognoter, *International Journal of Man-Machine Studies*, 34(2), 1991, 185-210.

[20] Vaghi, I., Greenhalgh, C., and Benford, S. (1999), Coping with Inconsistency due to Network Delays in Collaborative Virtual Environments. *Proc. ACM Workshop on Virtual Reality and Software Technology*, 1999, 42-49.

[21] Wu, J., and Ouhyoung, M., A 3D Tracking Experiment on Latency and Its Compensation Methods in Virtual Environments, *Proc. ACM Symposium on User Interface and Software Technology*, 1995, pp. 41-49.