Improving Awareness with Remote Control Point Movement Prediction in Real-Time Collaborative Graphics Editing Systems

Bo Jiang[1,2], Jiajun Bu[1], Chun Chen[1], and Jianxv Yang[1]

[1]College of Computer Science, Zhejiang University, Hangzhou, P.R.China
[2]College of Computer and Information Engineering, Zhejiang Gongshang Universtiy, hangzhou, P.R.China

Summary
Predicting real-time movement trace of remote objects or the control points of objects when delay-jitter occurs is a critical and challenging issue in most Internet-based collaborative graphics editing systems. This paper presents two novel algorithms: Machine Learning algorithm and Changeable Scale algorithm. Related experiments were carried out to test the effectiveness of the algorithms. Results show that the algorithms can improve the accuracy to restore the remote motion smoothly and the usability of the system can be greatly enhanced.

Key words:
CSCW, collaborative editing system, awareness, prediction, jitter

Introduction

Graphic editing systems (GES) [1, 2, 3, 4], which stand for a particular type of real-time collaborative editing application, enable geographically distributed users to edit the same graphics document simultaneously. Since each cooperative site maintains a replica of the shared graphics document for the sake of achieving high responsiveness, editing operations that issued by users should be broadcasted to synchronize the other scattered shared patterns. Providing collaborative users with sufficient awareness [5] information in GES assists cooperators understand others' up-to-the-moment activities in a shared workspace and it might assure successful collaboration in internet-based real-time GES.

Besides transmitting audio and video signals between cooperators, presenting the state of artifacts in a shared working setting is one of the most effective ways to provide awareness. Monitoring others' real-time drawing procedure, especially the remote moving tracks of pointers can greatly help users to understand and even predict others' designing intention. Therefore, presenting live variation procedure of remote graphics object is valuable for supporting collaboration in real time. However, networks may exhibit variability in delay, which is called jitter, can result in a jerky presentation of remote participant's actions in Internet environment. Affected by delay-jitter, the movement of objects or key control points could not be presented consecutively on remote sites, which may lead to misunderstanding and even operation conflicting in collaborative graphics design. Therefore, some related work has been done to ameliorate the effects of delay-jitter to display the remote motion in time and smoothly in order to achieve good awareness.

Tracing [6] the visual embodiments of groupware system enhances the visual representation of collaborators' motion and complements the problem of jitter. As tracing could hardly maintain the immediacy or the naturalness of the original embodiments' motion, the technique is limited. [5, 7] applied Dead-reckoning to improve player's interaction with distributed objects in games. However, Dead-reckoning only presents good performance in predict the motion of objects that force-based and strong inertial properties. [8] Dead-reckoning prediction was applied to reduce the effects of jitter on telepointer motion. Experiments were carried out and suggest that prediction can increase the immediacy and naturalness of remote interaction in groupware system. Yet the accuracy of prediction remains a problem.

In our work, motion prediction of control point of objects in GES was studied. Machine Learning Algorithm was presented to improve the predicting accuracy of Dead-reckoning by adjusting the prediction algorithm dynamically according to the former prediction error. And the performance was further enhanced by integrating Changeable Scale with the former algorithms. The control point movement would rarely be random because control point motion was bound with certain designing artifact. Experiments showed that the improved prediction algorithms maintain the performance of prediction accuracy at usable levels.

The structure of this paper is as follows: Section Two specifies the object-based collaborative GES and the control point based MOVE operations. Section Three depicts the jitter effect on Control Point Based MOVE Operation. Section Four presents two prediction algorithms and the corresponding experiments that test the effectiveness of the algorithms. Finally, Section Five concludes the paper.

Manuscript received March 25, 2006.
Manuscript revised March 30 , 2006.

2. Object-based GES and Control Point Related Movement

Graphics editing system can be classified into two categories: objected-based[8, 11, 13, 14] and bitmap-based[14, 15, 16, 17]. In bitmap-based systems, operations act directly on the drawing area but not on objects, and are generated by modifying the pixels' color in the drawing area. While in object-oriented systems, graphic objects such as rectangle, line, etc. can be created and modified. Each object is represented by attributes such as color, position and so on. Operations act on the objects and are generated by modifying the attributes of the objects [18]. Control point movement is a typical type of operations that can be acted on objects in object-based GES[4, 9]. The operation that related to control point movement is named CPB MOVE Op. In Fig. 1 operation MOVE changes the polygon's position from left-bottom to right-top. In Fig.2 the shape of the polygon is modified by dragging Point C. As it is shown in the following two figures, control points are the centre of the polygon in Fig. 1 and Point C in Fig. 2 respectively.

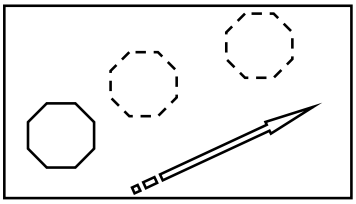

Fig.1. Move the polygon.

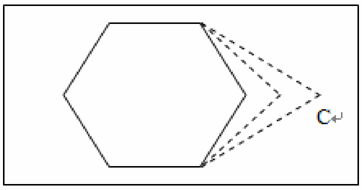

Fig. 2 Move the control point C of the polygon.

In real-time GES, replicated architecture is always adopted to achieve high responsiveness. Operations are broadcasted to remote cooperative sites as soon as it is issued by local user. CPB MOVE Ops that transmit the XY locations of control points are sent to assure that remote users can monitor the continuous object movement or variation. Tracing the remote CPB MOVE Ops, remote users may aware local user's actions and to predict his future drawing intension.

3. Jitter Effect

Distributed operation introduces communication delays between the collaborative sites on the Internet. Often, networks exhibit variability in delay, called jitter which can result in a jerky presentation of remote participant's actions. Continuous CPB MOVE Ops stream are sensitive to jitter that is due to the transmission over Internet, such as transmission delay, propagation delay and queuing delay. Delay-jitter may lead to semantic misunderstanding and operation conflicting as described as follows:

1.semantic misunderstanding: users may have an ambiguous idea of the semantic intention of remote user by monitoring the jerky display;

2.operation conflicting[8]: as it can be hardly to specify whether the halt of the motion is caused by jitter or not, users may take it for granted that the remote CPB MOVE Op task has been fulfilled and move the object that conflict with remote CPB MOVE Op.

Since jitter may make user have poor awareness on remote user's CPB MOVE Op, predicting the control point movement track while jitter occurs is necessary to promote the usability of GES.

4. Prediction Algorithms

Predicting the next location of control point based on past positions to simulate the actual track of the motion can eliminate the negative effect that brought by jitter. In the flowing paragraphs we present prediction algorithms used in our prototype system. To determine whether prediction is effective, we carried out corresponding experiments.

4.1 Machine Learning Algorithm

(1) Algorithm

Dead-reckoning has been widely and successfully used in internet game where Dead-reckoning improved player's interaction with distributed objects [7, 9, 10]. The prediction algorithm can be specified as follows:

We indicate the next position of the control point by (Xnext, Ynext) while the current position of the control point by ($X_{current}$, $Y_{current}$). The next position is calculated according to the following formula:

$X_{next} = X_{current} + aveVelocityX + aveAccelerationX$,

$Y_{next} = Y_{current} + aveVelocityY + aveAccelerationY$.

where,

$aveVelocity_X = (X_{last} - X_{flag}) / (T_{last} - T_{flag})$,

$aveVelocity_Y = (Y_{last} - Y_{flag}) / (T_{last} - T_{flag})$.

aveVelocityX and aveVelocityY indicate current average velocities of the control point in x and y axes. The average velocity is calculated through the last known position (X_{last}, Y_{last}) at the time of T_{last} and another position (X_{flag}, Y_{flag}) received some period before, at the time of T_{flag}.

$aveAccelerationX = (VelocityX_{last} - VelocityX_{flag}) / (T_{last} - T_{flag})$,

$aveAccelerationY = (VelocityY_{last} - VelocityY_{flag}) / (T_{last} - T_{flag})$.

aveAccelerationX and aveAccelerationY indicate current average acceleration of control point in x and y axes. The average acceleration is calculated through the last known velocity at the position (X_{last}, Y_{last}) at the time of T_{last} and another velocity at the position (X_{flag}, Y_{flag}) at the time of T_{flag}.

A new prediction system, which is called Machine Learning Algorithm, is able to adjust the prediction algorithm dynamically according to the former prediction error in the practical environment. The algorithm based on the last known position refines the prediction trace. The predicted position is calculated as follows:

$X'_{next} = X_{current} + aveVelocityX + aveAccelerationX + □x = X_{next} + □x$,

$Y'_{next} = Y_{current} + aveVelocityY + aveAccelerationY + □y = Y_{next} + □y$.

where,

is □x a variable to correct the value of X_{next} while □y is a variable to correct the value of Y_{next}. □x is related to: (1) the difference between last predicted X and first true X received at the end of jitter in the previous prediction process. (2) Current jitter lapse. (3) Jitter period in the previous prediction process. If it is the first time to prediction, we did not consider □x and □y .Calculation of is similar to that of X. To illustrate the algorithm:

$2 = (X_1'_{last} – X_{1\ true}) * JitterLapse_2 * 0.5 / JitterPeriod_1$,

where,

subscript 1 indicates the previous prediction process, while subscript 2 indicates the current prediction process. $X_1'_{last}$ is the last predicted position in the previous prediction process. X_1 true is the blocked last true value by the previous network jitter in the previous prediction process.

(2) Effectiveness

To test the effectiveness of the Machine Learning Algorithm, we made an experiment. We carried out an experiment to test the effectiveness of the scheme. Ten volunteers (5 male, 5 female) from local University and pattern designing company were invited to our lab. All volunteers were right handed and were frequent users of mouse and windows systems (at least 40 hours/week). Six of ten (3 male, 3 female) are students of computer major, four of ten (2 male, 2 female) are pattern designing professionals. The experiment was conducted on Dell PC running CoDesign system application, using a 17-inch monitor set to 1024x768 resolution, 256M memory and 2.4G CPU.

While volunteers were drawing and moving some graphics objects from one position to another, jitters were generated by a simulation application to simulate unstable network. Our prediction system application adopts the Dead-reckoning prediction and Machine Learning algorithm respectively. At the end of jitter, system calculates the error of prediction, difference between the last predicted position and corresponding true position extracted from received package at the end of jitter.

The testing result is shown in Fig.3.

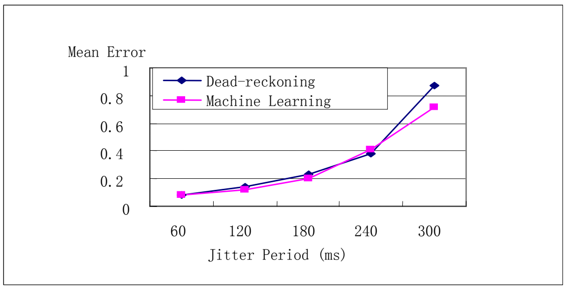

Fig. 3. Mean Error for 5 jitter periods with Machine Learning. The mean error is a difference divided by a unit length. If the mean error > 1, we record it as 1, for it is beyond the prediction region too far. The difference is distance between the last predicted position and first position extracted from received package at the end of jitter. The unit length indicates a local area with user's most frequent activities. The area is centered by user's current position with a radius of the unit length. The unit length is defined according to system and application. In our prediction system, the unit length = 10% of diagonal of canvas.

As it is shown in Figure 3, Machine Learning is better than Dead-reckoning in terms of prediction accuracy, especially at the higher jitter periods. However, as it is illustrated in Fig. 3, at certain jitter periods such as 240ms the effectiveness of Dead-reckoning is better than that of Machine Learning.

4.3 Changeable Scale Algorithm

(1) Algorithm

As we know, in a certain period of time user usually concentrates on his/her nearby editing areas, which is called current active working area. When a control point is moved to somewhere else, new position will not be far from the original one. The improved prediction algorithm, Changeable Scale, adapts to human graphics editing habit in order to enhance the accuracy of the prediction in GES.

The main idea of the algorithm this time is to adjust prediction power according to the distance between the position user begin to drag and predicted position by original algorithm, such as dead-reckoning or machine learning. The longer the distance between the two positions is, the lower the next prediction power could be. We describe the algorithm as follow:

Where, α , the prediction power, is determined mainly by the distance between the two positions. Multiple factors could have contributed to the construction of α , here we just simplify it to the following form:

$$\begin{cases} X'_{next} = X_{next} * \alpha \\ Y'_{next} = Y_{next} * \alpha \end{cases}, where \begin{cases} X_{next} = X_{current} + aveVelocity_X + aveAcceleration_X \\ Y_{next} = Y_{current} + aveVelocity_Y + aveAcceleration_Y \end{cases}.$$

,

$$\alpha = \begin{cases} 1 & , Length \le UnitLength \\ UnitLength / Length & , Length > UnitLength \end{cases}$$

where,

Length is the distance between the position user begin to drag and predicted position by original algorithm. UnitLength is a unit length defined by the system as described above.

Therefore, there is a circle centered by the position user begin to drag, with a radius of UnitLength. If the predicted position by original algorithm is within the circle, we take the dead-reckoning algorithm unchanged for we get $\alpha = 1$. While the predicted position by original algorithm is without the circle, the prediction power has to work taking the value of UnitLength / Length.

(2) Effectiveness

According to the experiment, comparison of the prediction with changeable scale to the prediction without changeable scale is shown in Fig. 4.

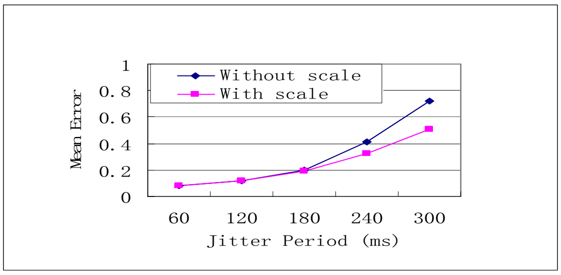

Fig. 4. Mean Error for 5 jitter periods with Machine Learning with and without changeable scale.

As shown in Fig. 4, prediction with changeable scale is better than that without changeable scale in terms of prediction accuracy, especially when jitter period increases.

5. Conclusions

The main objective of our work is to restore the consistent and immediate motion track of artifacts that controlled by remote user to promote collaborative awareness in real-time GES. Our solution to the problem of jumpy presentation of remote action track is predicting the trace of control points while jitter occurs. Machine Learning algorithm improves prediction accuracy by taking the former prediction error into account. Changeable Scale algorithm which adapts to users' editing habits presents better prediction performance. Related experiments were explored to report and compare the effectiveness of each algorithm. It can be shown that the algorithms we proposed work correctly in GES, the impact of jitter is ameliorated and the degree of collaborative awareness is greatly improved.

Acknowledgments

This paper is supported by National Natural Science Foundation of China (60573176) and Zhejiang Provincial Natural Science Foundation of China under Grant No. Z603231.

References

[1] C. Sun and D. Chen, Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems. ACM Transactions on Computer-Human Interaction. 9,1 (2002): p. 1-41.

[2] X. Wang, et al. Achieving Undo in Bitmap-Based Collaborative Graphics Editing Systems. In Proceedings of 2002 ACM Conference on Computer Supported

Cooperative Work (CSCW'02), November 16-20. New Orleans, Louisiana, USA.

[3] Xianghua Xu, Chun Chen, Jiajun Bu and Yong Li, Distributed Dynamic-locking in Real-time Collaborative Editing Systems. In Proceedings of 10th International Conference on Groupware, September 5-9, 2004. Springer-Verlag LNCS3198. p.271-279.

[4] Bo Jiang, Chun Chen, Jiajun Bu, CoDesign-A collaborative pattern design system based on agent. Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design, Canada, (2001) 319-323.

[5] Gutwin, C., and Greenberg, S. (2004) The Importance of Awareness for Team Cognition in Distributed Collaboration. In E. Salas and S. M. Fiore (Editors) Team Cognition: Understanding the Factors that Drive Process and Performance, 177-201, Washington:APA Press.

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

[7] Gutwin, C., Dyck, J., Burkitt, J. Using Cursor Prediction to Smooth Telepointer Jitter. Proc. ACM Group 2003, 294-301.

[8] Sun, C. and Chen, D. A Multi-version Approach to Conflict Resolution in Distributed Groupware Systems. In Proceedings of the 20th IEEE International Conference on Distributed Computing Systems. Taipei, Taiwan. April 2000. 316-325.

[9] 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.

[10] 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.

[11] D. Li and R. R. Muntz. COCA: Collaborative Objects Coordination Architecture. In Proceedings of ACM CSCW'98. Seattle, USA. Nov. 1998. 179-188.

[12] M. Roseman and S. Greenberg. Building real time groupware with GroupKit, a groupware toolkit. ACM Transactions on CHI, March 1996.

[13] N.A. Streitz, J. Geiler, J.M. Haake, and J. Hol. DOLPHIN: Integrated meeting support across liveboards, local and remote desktop environments. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94), Chapel Hill, N.C., October 1994. 345--358.

[14] A. BEGEL. CrystalBoard. CS294-7 "CSCW using CSCW" Course Project Report. December 1997.

[15] B. Wilson. WSCRAWL 2.0: A Shared whiteboard based on X-Windows. In Designing Groupware for Real Time Drawing. Editors: S. Greenberg, S. Hayne and R. Rada, McGraw Hill, 1994.

[16] C. Wolf, J. Rhyne, L. Zorman. WE-MET (Window Environment-Meeting Enhancement Tools). In Proceedings of CHI'91. New Orleans, LA, USA. 1991. 441-442.

[17] S. Greenberg, and R. Bohnet. GroupSketch: A Multi-User Sketchpad for Geographically Distributed Small Groups. In Proceedings of Graphical Interface'91. Calgary, Alberta, CANADA. Jun. 1991. 207-215.

[18] D. Chen and C. Sun. A Distributed Algorithm for Graphic Objects Replication in Real-time Group Editors. ACM GROUP'99, Phoenix Arizona, USA, 1999. 121-130.

Bo Jiang received the M.S. degrees in Computer Science from Zhejiang University in 2002. She stayed in Ubiquitous Computing Research Laboratory (UCRL) Zhejiang University to study CSCW, and Ubiquitous Computing since 2000. She is now a PhD candidate in Zhejiang University.

Jiajun Bu received the PhD degrees in Computer Science from Zhejiang University in 2000. He is now an associate professor in College of Computer Science of Zhejiang University. His research interests include CSCW, Ubiquitous Computing, and Computer Vision.

Chun Chen is now a professor in College of Computer Science of Zhejiang University. His research interests include CSCW, Ubiquitous Computing, Computer Vision, and CAD/CAM.

Jianxv Yang received the B.S. and M.S. degrees in Computer Science from Zhejiang University in 2004 and 2006, respectively. During 2004-2006, he stayed in Ubiquitous Computing Research Laboratory (UCRL), Zhejiang University to study CSCW, Search Engine and Ubiquitous Computing.