Remote Move-Related Operation Awareness in Real-Time Internet-Based Collaborative Graphics Design Systems

Bo Jiang[1,2], Jianxv Yang[2]
[1]College of Computer Science and Information Engineering,
Zhejiang Gongshang University, Hangzhou, P.R.China
[2]College of Computer Science, Zhejiang University, Hangzhou, P.R.China
nancybjiang@zju.edu.cn; yangjianxv@yahoo.com.cn

Abstract

Supporting awareness of other cooperators is an idea that holds promise for improving the usability of real-time distributed collaborative graphics design systems. However, network jitter may cause serious problems in providing real-time awareness information such as monitoring the remote motion of objects or the control points of objects in such Internet-based designing system. Halting and jumping presence of remote move-related operations may occur. In this paper, we present novel algorithm and scheme that can improve the move-related operation display performance of accuracy and smoothness. Experiments were carried out to test the effectiveness of the scheme and the algorithm. The results show that by applying effective remote motion prediction and introducing jitter buffer, the usability of the system can be greatly enhanced.

Keywords: Operation awareness, Collaborative graphics design, Prediction.

1. Introduction

The goal of real-time collaborative graphics design systems (CGD) [1, 2, 3] is to allow people that located in different places to design a pattern together, as naturally as they do face-to-face in the same physical location. CGD can be classified into three categories: bitmap-based [4, 5], object-based [6, 7] and hybrid [2]. In bitmap-based systems, operations act directly by modifying pixels' color in the drawing area. In object-based systems, graphic objects such as rectangle, line, etc. can be created, modified and deleted. Each object is represented by attributes such as position, color, etc. Operations act on objects, and are generated by modifying the attributes of the objects. In this paper we mainly focus on the object-based systems.

To achieve high responsiveness, a full replicated architecture is adopted in our prototype system – CoDesign. In replicated architecture, editing operations are executed on local machine and issue to other remote sites, and each cooperative site keeps the copy of the shared documents.

In typical object-based CGD systems, among various types of operations, operations related to move are most valuable to be monitored by remote users. Move-related operations can either be simple move object operation but also be modify object by changing the position of a control point of the object like modifying the shape of a polygon. Since monitoring the real-time process of remote move-related operations can assist cooperators to better interpret the other people's activity, anticipate their actions and interpret their gesture [8], presenting the operating procedure of these operations are usually applied in groupware systems to enhance the ability of remote awareness.

However, jitter that caused by network delay will make negative influence on displaying the procedure of remote moving or dragging operations on certain artifacts. Jitter, which is the variation in latency over internet, will always leads to jumpy display of such operations. People feel that it is hard to understand and predict others' editing intension. Besides, operation conflicting in collaborative graphics design may occur in certain cases. Therefore, it is very important to ameliorate the effect of delay-jitter to display the remote move-related operation in time and smoothly.

In this paper, we first introduce a jitter buffer scheme that can make display of remote move-related operation more continuously. Then a prediction algorithm is presented to use the predicted coordinate to mimic the real one that is not received because of delay-jitter. Finally, the prediction algorithm is integrated with jitter buffer scheme to achieve better performance of displaying the process of move-related operations and thus provides cooperators with better awareness information in CGD.

The rest of this paper is structured as follows. Section Two introduces the concept of move-related operations and analyzes the problem that jitter brought to such kind of operations. Section Three presents the scheme of jitter buffer in CoDesign. Section Four proposes a prediction algorithm and the fusion of the prediction algorithm with jitter buffer scheme. Section Five compares our scheme and algorithm to the related work. Finally, Section Six concludes the paper.

2. Jitter Effect on Presenting MOVE-Related Operations

As it is known that MOVE-related operations depends on the movement of certain control point of the corresponding object. As it is shown in Figure 1 and Figure 2, control point can be the center of the object in move operation, and it can be one of the vertexes of the object in modify operation.

To present the real-time process of remote move-related operations, it is always natural to send the XY locations of the control point of the object. The control point position should be updated at a high enough frequency, and besides, the update frequency must remain consistent to ensure that the pacing of the movement closely matches the original motion of the remote action for users to gain good awareness.

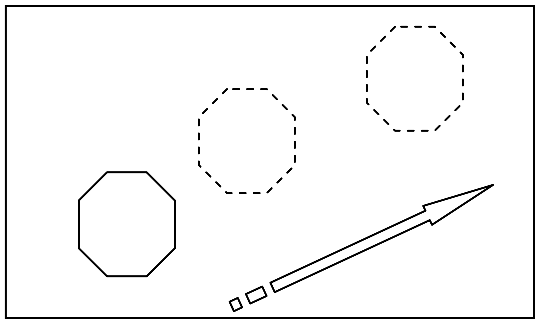

Fig. 1. The control point is the center of the polygon in move operation. The arrow shows the moving direction.

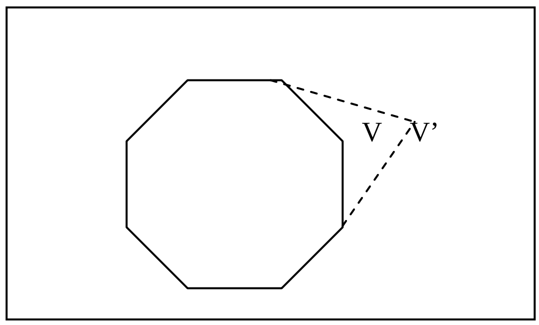

Fig. 2. The control point is the vertex V of the polygon in modify operation.

However, in Internet there is variation in latency, which we called jitter, due to changing network traffic conditions. This causes a remote user's operations to appear jerky, which makes it difficult for users to interpret and predict the remote user's action.

We carried out an experiment to test the effect of jitter on presenting move-related operation. 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, using a 17-inch monitor set to 1024x768 resolution, 256M memory and 2.4G CPU.

Random curves were created automatically, and jitters were generated by a simulation application to simulate unstable network. The curves were presented on users' screen gradually according to the delay of jitter. Users were demanded to click the predicted point on the curve before it was shown.

It is shown in Figure 3 that to viewers the negative effect on the accuracy of users' point prediction that brought by jitter become more serious when jitter is more than 200ms.

It is obvious that with the jitter delays increasing users find it more difficult to predict what remote user wants to design and which leads to misunderstanding of others' design intension. As a result, design intension conflict and operation conflict may occur in certain cases.

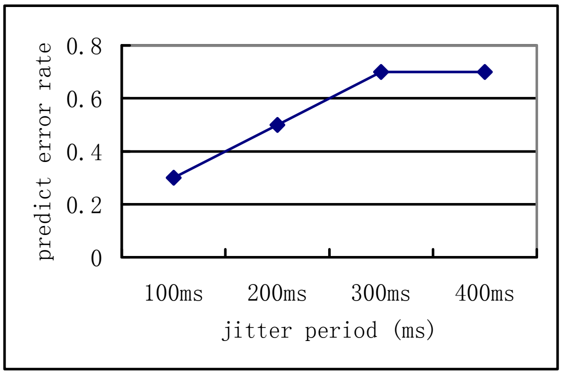

Fig. 3. Jitter effects on predicting curves.

3. Jitter Buffer Scheme

Jitter has primarily been studied in streaming media, where very small variations in playback of audio and video files can be noticed by people. To remove jitter requires collecting enough multimedia data packets from a channel in a buffer so that the packets can be played out with a constant inter-packet gap corresponding to the codec interval.

This kind of jitter buffer scheme that can smooth the display of stream media may also be applied in presenting remote MOVE-related operations. UDP, a lightweight transport protocol, is used to send awareness information like the process of the motion of the object that related to a certain operation in order to reduce network delay. On the contrary, the control information and the result of an edit operation should be transmitted via reliable network protocol like TCP. Therefore, the awareness messages of the operation process can be stored in a jitter buffer before they are displayed until the end of the MOVE-related operation signal is received via TCP. According to the above primary jitter buffer scheme, remote MOVE-related operations are assured to be presented continuously. However, the buffer scheme also makes negative effect on awareness for the display is delayed especially when the process of the MOVE-related operation lasts for a long period of time.

Since real-time awareness is crucial for a successful CGD, the jitter buffer scheme is improved. After a packet is arrived, it is replayed after a certain period of time to ameliorate the jitter effect and also avoid the delayed display problem that exists in the primary jitter buffer scheme. To decide if a packet should be played or stored in the jitter buffer, the interval of displaying two packets should be set. According to Section two, the accuracy of users' point prediction that brought by jitter becomes more serious when jitter is more than 200ms. The interval of displaying two packets which are buffered is set no more than 200ms. As it is shown in Figure 4, the interval is 200ms. The interval can be adjusted to a proper value according to the status of network.

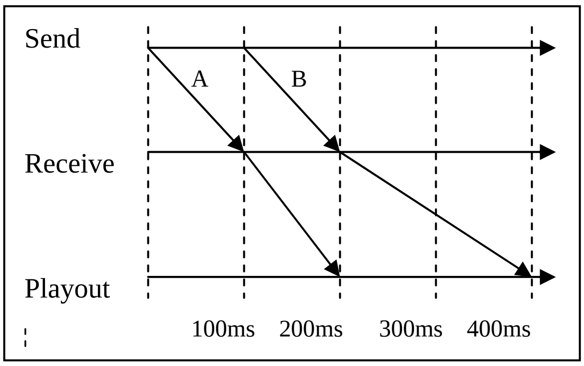

Fig. 4. Improved jitter buffer scheme. In this case, the jitter period between packet A and packet B is 100ms. The display interval is 200ms.

Introducing display interval into jitter buffer scheme, the overall performance of smoothness of displaying remote MOVE-related operation is improved.

Similar experiment (like that addressed in Section two) was carried out to test the effectiveness of improved jitter buffer scheme. The testing result is shown in Figure 5.

With jitter buffer scheme, packets are placed into proper time sequential order, applying a specified jitter delay, and playing them back at the proper rate (with a certain display interval). However, as shown in Figure 3, the predict error rate is high when the average jitter period is more than 300ms. In the following section, we present the scheme of control point prediction that can better complement the effect of jitter.

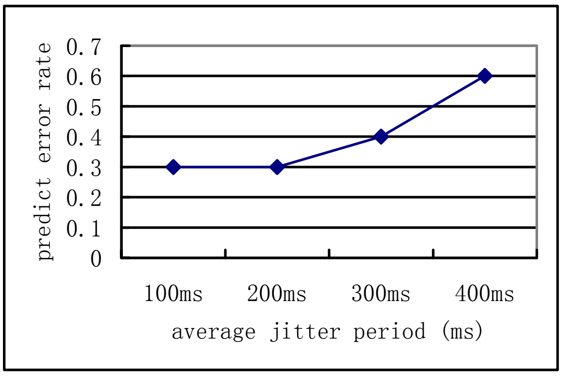

Fig. 5. Jitter effects on predicting curves by applying improved jitter buffer scheme. The average jitter period is set as the figure. The display interval is 200ms.

4. Motion Prediction Scheme

Presenting the mimic track of actual process of remote MOVE-related operation may improve the visual effect of collaborative users and thus achieve perfect cooperation with high efficiency. To simulate the actual track and eliminate the negative effect of jitter, an algorithm that can predict the next location of control point based on past positions is designed. In the flowing paragraphs the prediction algorithm used in our prototype system is presented. To determine whether prediction is effective, we carried out the corresponding experiment.

As it is shown in Figure 3 and Figure 5, error rate of prediction is unbearable with the increase of network jitter period. To solve the problem, we present an algorithm that is able to predict the next control point and adjust the prediction algorithm dynamically according to the former prediction error. The algorithm is based on the last known position. The next predicted position is calculated as follows:

X' = X + aveVX + aveAccX + DistX = X' + DistX ,

Y' = Y + aveVY + aveAccY + DistY = Y' +DistY.

where,

X' and Y' is the predicted coordinate of next point, while X and Y is the coordinate of current point. DistX is a variable to correct the value of X' while DistY is a variable to correct the value of Y'. DistX 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 DistX and DistY. Calculation of DistY is similar to that of DistX.

To illustrate the algorithm:

$DistX_2 = (X_1'_{last} - X_{1\ true}) * JitterLapse_2 * 0.5 / JitterPeriod_1$,

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.

To test the effectiveness of the prediction algorithm, an experiment was carried out. 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 adopted the prediction algorithm. At the end of jitter, system calculated the error of prediction, difference between the last predicted position and corresponding true position was extracted from received package at the end of jitter. The testing result is shown in Figure 6. When jitter period is larger than 300ms, mean error increases. We made another experiment that integrates the jitter buffer scheme and prediction. When jitter

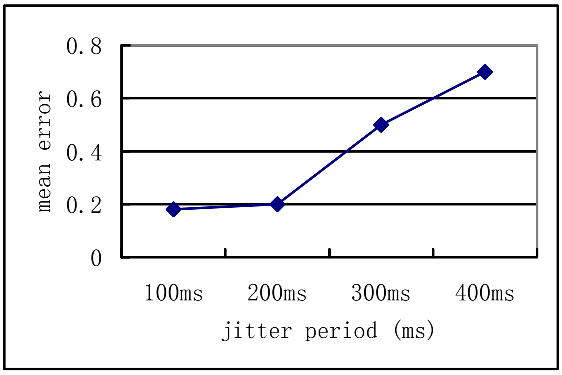

Fig. 6. Mean Error for 4 jitter periods with prediction algorithm. The mean error is a difference divided by a unit length. If the mean error > 1, we record it as 1, for it is beyond the prediction region too far. The difference is distance between the last predicted position and first position extracted 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.

period was larger than display interval, a predicted value of the point was used to display. Figure 7 shows the effectiveness of this integrated scheme. The result indicates that the effectiveness of integrated scheme is better than the former ones especially when jitter period is more than 200ms.

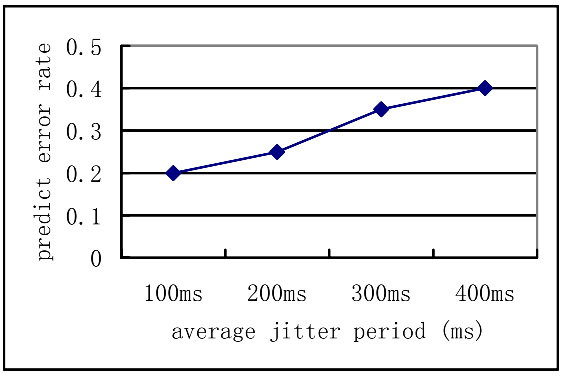

Fig. 7. Effectiveness of the scheme integrated jitter buffer scheme and prediction algorithm.

5. Comparison to Related Work

Jitter has primarily been studied in streaming media, and the multimedia packets are usually played out with a constant inter-packet gap corresponding to the codec interval. Yet unlike display MOVE-related operations, buffer scheme applied to displaying streaming media tolerates to the delay of display.

Some work has been done in eliminating the jitter effect on smoothing the display of remote motion of embodiments. In [9, 10], Dead-reckoning is used to improve player's interaction with distributed objects in games. However, Dead-reckoning only fits for predicting the motion of objects that are force-based and strong inertial properties.

In our work, motion prediction of MOVE-related operations in CGD has been studied. And by integrate the prediction algorithm with jitter buffer scheme, users are able to monitor the process of remote MOVE-related operation track more accurately and smoothly.

6. Conclusions

Providing real-time awareness information smoothly to cooperators is crucial in developing collaborative graphics design systems. Although MOVE-related operation is one of the most important kinds of awareness information, little work has been done on the display of the process of such remote operation. In this paper, we present a jitter buffer scheme that sets display interval between two buffered packets. Moreover, a prediction algorithm that predicts the next coordinate of the control point on the track of MOVE-related operations is used to imitate the real track. And the prediction algorithm that integrate with jitter buffer scheme shows better effectiveness of accuracy and smoothness in displaying remote MOVE-related operations.

We are pursuing our work to improve the motion prediction algorithm to achieve better performance.

7. Acknowledgements

This paper is supported by National Natural

Science Foundation of China under Grant No. 60573176 and Zhejiang Provincial Natural Science Foundation of China under Grant No. Z603231. Thanks to Dr. Yong Li for his advices, and to the volunteers for taking part in the experiments.

8. References

[1] 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, p319-323, 2001.7.

[2] 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.

[3] 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.

[4] Begel, A. CrystalBoard. CS294-7 "CSCW using CSCW" Course Project Report. December 1997.

[5] 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.

[6] Boltman, A. Children's Storytelling Technologies: Differences in Elaboration and Recall. HCIL-2001-24, CS-TR-4305, UMIACS-TR-2001-82. Ph.D. Dissertation. October 2001.

[7] 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.

[8] Gutwin, C. (2002) Traces: Visualizing the Immediate Past to Support Group Interaction. In Proceedings of the Conference on Human-Computer Interaction and Computer Graphics (GI'02), Calgary, Morgan Kaufman, 43-50.

[9] 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.

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