

Exploring Visual Feedback of Change Conflict in a Distributed 3D Environment

Mark S. Hancock^{1,2}

John David Miller¹

Saul Greenberg²

Sheelagh Carpendale²

¹Intel Corporation
2111 NE 25th Ave
Hillsboro, OR 97124

john.david.miller@intel.com

²University of Calgary
2500 University Drive NW
Calgary, AB, Canada T2N 1N4

{msh,saul,sheelagh}@cs.ucalgary.com

ABSTRACT

Teams that are geographically distributed often share information both in real-time and asynchronously. When such sharing is through groupware, change conflicts can arise when people pursue parallel and competing actions on the same information. This leads to problems in how the systems and its users maintain a consistent view of shared information across distance and time. We explore change awareness of conflicts in a three-dimensional distributed shared space. Our user study compares the use of visual feedback to an optimistic concurrency control strategy for both synchronous and asynchronous distributed groupware. Our feedback provides a means for synchronous users to recognize and resolve real-time changes, and for asynchronous users to view and resolve changes when switching from an offline to online mode of work. Results of our study suggest that the visual feedback serves as a useful feedthrough mechanism in the synchronous case, but that asynchronous users may be overwhelmed by the quantity of changes if they come online after many changes have been made.

Categories and Subject Descriptors

H5.3. [Information interfaces and presentation (e.g., HCI)]: Group and Organization Interfaces – Computer-supported cooperative work.

General Terms

Design, Human Factors.

Keywords

Change conflict, divergence, distributed collaboration, synchronous, asynchronous, visual feedback.

1. INTRODUCTION

We approach the problem of change conflict resolution in a 3D shared information space by introducing *stretchies*, a metaphor that combines visual feedback with the ability to easily resolve conflicts. In this paper, we describe stretchies and a user study designed to compare synchronous and asynchronous use of this visual feedback to a typical systems-based approach.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI '06, May 23-26, 2006, Venezia, Italy.

Copyright 2006 ACM 1-59593-353-0/06/0005...\$5.00.

Finding workable solutions for change conflict resolution is growing in importance because, due to the increase in available technology and the ubiquity of network connectivity, people are increasingly able to work away from the office. In fact, it is not uncommon for teams to include members in different cities, time zones, countries and even continents. With this shift comes the need to support teams distributed across both distance and time.

When teams are co-located, they often share and manipulate information in some common area, such as a whiteboard, a table, or even an entire room. When distributed, teams also often require some common repository of information, which all members can access at any time, whenever they want and from wherever they want. As in the co-located case, this repository acts as a shared memory for the team, and can be both an information storage medium and a communication medium. For example, team members can store commonly authored documents, references, and schedules. Teams can also use the repository to present information to one another or to collaboratively complete a task.

In order to support this shared information space, all team members need a consistent view of the information. However, inconsistencies can arise through *change conflicts*, which occur when two or more users concurrently make changes to unsynchronized data. A common approach to dealing with change conflicts is to either impose some constraint on the user such as locking the data when it is in use or to automatically synchronize the data with computer algorithms. However, these algorithmic approaches cannot be sensitive to all the group's needs [10].

Stretchies leverages previous work [8,9] in that they provide the user with the flexibility to work opportunistically in either an asynchronous or synchronous fashion. Our results suggest that this visual feedback is sufficient for synchronous users, but that asynchronous users require more information and more control of how that information is presented. More generally, our results indicate that, despite the similarity in the theory which describes the change conflicts in both cases, the needs of asynchronous users vary greatly from those of synchronous users.

2. RELATED WORK

Stretchies, the visual feedback we provide for change conflict in a 3D shared workspace, leverage previous literature from several research areas: the arguments made for 3D interfaces; conflict, divergence in groupware; change awareness feedback management in distributed collaborative environments; and 2D visual representations that use a 'stretch' metaphor to connect related objects.

2.1 3D Groupware Interfaces

Our choice to tackle the problem of providing appropriate feedback for change conflicts in a 3D groupware system reflects our belief that this genre is worthy of study, particularly for groupware. An important factor that influenced this decision is that it appears that 3D interfaces may already be the most popular basis for groupware. That is, the heavy use of the many 3D virtual worlds and Internet games suggests that there may be more users of 3D groupware than of 2D groupware image and text editors. In spite of this, we are aware that the issue of 3D vs 2D interfaces is in active debate in the research community [3,4,5,15,18,19]. For instance, studies exist that indicate spatial advantages when using 3D projections with motion or stereo [19]. Other studies question these results [3,4,5]. Pekkola [15] debates the value of 3D in collaborative virtual environments. In particular, we selected Miramar as an example of a 3D information management interface that has had positive feedback through user studies [14] (see Section 3 for details about the Miramar interface). Our focus is on how change conflicts can be visualized in a 3D virtual collaborative workspace, and as we will see, our method exploits several properties typically associated with 3D worlds to improve users' ability to visually comprehend object changes in the workspace: animation, spatial positioning, and changes of viewpoint. While it is possible that games may exist that implement some mechanism for visualizing change conflicts in 3D, to our knowledge there has been no serious study of any method.

2.2 Conflict and Divergence

Conflict occurs when people pursue parallel and competing actions on the same object. Divergence happens when the system allows those actions to continue, thus creating multiple outcomes that must be resolved at some point.

Normally, conflict is managed in synchronous groupware systems by one of the many concurrency control mechanisms adapted from the distributed systems field. Greenberg and Marwood [10] provide an overview of the benefits and drawbacks of specific concurrency control mechanisms in Computer-Supported Cooperative Work (CSCW) applications, especially the tradeoffs inherent when choosing between optimistic and non-optimistic policies for managing serialization and locking. Their critical point is that groupware solutions to concurrency control are inseparable from user interface design: the method chosen can have a profound effect on how the end user perceives the interaction. For example, serialization schemes may guarantee consistent order, but may not fulfill the intentions of participants' actions. Optimistic schemes introduce problems of how the interface can roll back to prior states when conflict occurs. Non-optimistic locking strategies introduce delays, added interaction effort, and reduce the possibilities for simultaneous interactions. Groupware-specific solutions to concurrency control are an active research area [16].

Dourish [8,9] introduces the notion of *multi-synchronous work*. He explains concurrency control algorithmically as a divergence-synchronization loop. That is, when multiple users are working in a distributed environment, each with the ability to modify the distributed data, conflicting changes result in divergence. These divergent states are then synchronized to maintain consistency among collaborators. In *synchronous work*, the frequency of synchronization is high (ideally, real-time, though in practice is

limited by bandwidth). In *asynchronous work*, the frequency of synchronization is low. Multi-synchronous work is thus a combination of both synchronous and asynchronous work within the same team. As seen shortly in this paper, we recognize the multi-synchronous nature of group work by studying its extremes: how people view the consequences of divergence in synchronous and asynchronous actions via a change awareness mechanism.

2.3 Workspace Awareness/Change Awareness

Gutwin and Greenberg [11] provide a descriptive theory for workspace awareness, the up-to-the-moment understanding of another person's interaction with a shared workspace. While the framework contains many elements (based around the *who*, *what*, *where*, *when*, *why* and *how* of changes that are made), the element most relevant to our work is awareness of other people's actions on artifacts. In particular, this awareness can help participants' coordination of actions and activities, where the right thing will happen in the right order and at the right time. Appropriate awareness may not only help people anticipate and avoid conflicting actions, but may also help them rapidly detect and repair conflicts when they do occur.

Tam and Greenberg [17] extend this framework to asynchronous work. They describe what information could be needed by a person trying to understand how a visual workspace used by others has changed. They argue that it is very difficult for people to identify changes if they rely only on memory ('can I recall what has changed since I last looked?'). Even when before and after scenes are placed side by side, differences – especially small ones – are very difficult to detect. Consequently they argue that asynchronous groupware systems need to include mechanisms that let people truly understand what has changed. They use this framework to critique several visualizations of change awareness information in an asynchronous distributed 2D furniture layout application [17]. This application visually identifies many elements of the change, including who made a change (with colour), the amount of changes made (with "change bars"), where and what change was made (with visual makers, such as arrows), and how a change was made (with ghosting and outlining). User-controlled animations (via a slider) are also used to play back the history of changes made. Their critique points out that these inventions have serious flaws: none are tested in actual use.

2.4 Stretching in 2D

Our visual feedback shows divergence by making a copy of the object in question and connecting the copies by stretched lines.

This idea of stretching has been used in many interfaces to indicate a connection between two-dimensional objects, such as icons. Drag-and-Pop and Drag-and-Pick [1,6] both use a stretched 2D visualization to represent a connection between an icon and a proxy of that icon. CoR²Ds [13] uses similar visual feedback to represent a connection between two 2D objects on the surface of a tabletop display. These visualizations differ from our stretchy visual feedback in that they do not represent two possible states on different machines, nor do they deal with synchronization of icons in a distributed environment, nor are they refined to 3D.

In the remainder of this paper, we introduce a visual feedback mechanism, called *stretchies*, showing positional change information in Miramar, a 3D shared workspace. In our study, we focus on this type of change in order to isolate our findings to the differences between synchronous and asynchronous use of this



Figure 1. A close-up view of a document in Miramar. The navigation bar at the bottom-left of the scene includes buttons for (from left) home, overview, and movement arrows. Shadows appear as a reflection of the object.

visual feedback. To further simplify the experiment, we also limit the information provided through the stretchies, and do not include other feedback, such as who, when and why the changes were made [17]. We compare the use of this feedback to an optimistic synchronization policy.

3. Miramar

Miramar marries a 3D workspace with distributed collaboration. As a 3D workspace, Miramar also allows users to “push back” applications from their 2D desktop into the 3D scene and arrange them through position and rotation in a 3D environment (see Figure 1). This 3D environment can be shared between multiple users both synchronously and asynchronously. The combination of a 3D workspace and distributed collaboration supports teams of users separated by both distance and time. Such teams can easily share information when they use the 3D workspace to facilitate shared knowledge and communication of work progress.

3.1 3D Environment

Miramar’s design considers three important factors in 3D environments: navigation, manipulation and presentation.

3.1.1 Navigation

In order to explore a 3D environment, it should be easy to navigate using a mouse and keyboard. Miramar uses several interaction mechanisms to facilitate navigation. Home and overview buttons allow the user to toggle between the current view and either the starting camera position or an overview perspective, respectively. These constant camera positions allow the user to easily navigate to a familiar viewpoint and prevent the user from “getting lost”. A user can also zoom in to “get a good look” at any object in the scene by clicking it and pressing the space bar. This feature allows users to navigate object-by-object without needing to search the scene to find an appropriate view. Users can also freely navigate with either the arrow keys or by clicking anywhere in the scene and dragging the mouse. A series of usability studies [reference removed for blind review] were used to iteratively design these navigation techniques, which were shown to be understandable and easy to use.

3.1.2 Manipulation

Users can place documents and Microsoft® Windows applications in the Miramar 3D environment. All such applications are adorned with a ‘sendback’ button which can be used to push the

window into the 3D scene by creating a 3D representation of the (2D) Windows object. Once in the scene, dragging an application with the mouse moves it along the x-z plane in the 3D environment. Objects can also be rotated and moved along the z-axis using modifier keys. Once in the scene, objects can be linked together, placed on “billboards” or otherwise organized in the 3D environment.

3.1.3 Presentation

Miramar provides several cues to help the user understand the three-dimensional objects in the scene. Animations indicate actions in the scene, such as the addition of an application or movement of the viewpoint (either through navigation or through the home/overview buttons). When objects are moved in real-time by other users, this movement is also animated. Shadows appear underneath each object to help users recognize the 3D position of objects. These shadows are not physically based, but are more like reflections in that they provide a two-dimensional copy of the object on the ground plane of the scene (see Figure 1). Thus, the top-down overview provides a 2D representation of the 3D elements in the scene.

3.2 Virtual Collaboration in Miramar

We have extended Miramar by making it possible for two or more users to open the same scene from more than one site, either synchronously or asynchronously. Each user can navigate in the scene independently of the other users, but modifications to the scene, such as additions, movement of objects, and deletions will be shown in the scenes of all users. This update will occur in real time (as far as bandwidth will allow) for all connected users. The scenes of any offline users are updated upon reconnection.

Miramar provides a repository of shared work for a team of users distributed across both distance and time. Users separated by distance only (e.g. within the same time zone) can use the same scene *synchronously* and users separated by distance and time (e.g. across multiple time zones) can work *asynchronously*.

3.3 Change Conflicts

When two or more copies of data exist on multiple computers distributed across a network, synchronization of these copies requires one computer to pass a message to the other (possibly through a server) whenever the state is changed. When two or more users are independently making changes to the same data set, conflicts can arise. For example, if two users (*User A* and *User B*) begin with a data set in the same state, S_0 , and *User A* changes the data from S_0 to S_a , but *User B* changes the data from S_0 to S_b before receiving the message from *User A*, two conflicting states arise, S_a and S_b (see Figure 2).

Changes may also be represented at many different levels. The state that is being changed could be chosen to be an entire Miramar scene, an object in the scene, or the contents of an object in the scene. Thus, a change graph can be used to represent the state changes at any or all of these different levels.

3.3.1 Conflict Resolution

One method of resolving conflicts is an *optimistic policy*, which assumes that conflicts are not common and/or that conflicts will not be harmful. Following the above example, *User A* would be in state S_b and *User B* would be in state S_a . If either user made another change, it would overwrite the state on the other machine. This policy is typical in many CSCW systems, such as a wiki.

Another method to handle change conflicts, a *pessimistic policy*, is to preempt them. This policy disallows the example given above. Instead, in order for either User A or User B to change from state S_0 , he/she must obtain a lock on the data before making the change. Whichever user first requests a lock is then allowed to make a change and the other user is prevented from doing so. This policy is common in many real-time groupware systems and in asynchronous source control systems such as Concurrent Versioning System (CVS) and Microsoft Visual SourceSafe.

A third method of handling such conflicts is to notify the users that a conflict has occurred and require that *they* resolve the conflict. This *user-controlled policy* is often used in systems that support multiple authoring of documents.

For further discussion of policies to handle change conflict, see Greenberg and Marwood [10].

When two or more users are simultaneously working in the same Miramar scene, *synchronous* change conflicts arise when both users attempt to make a change to the same object at the same time. More precisely, due to delays in passing messages across a network, two users can be in conflict if they both make a change in a conflicting time interval (see Figure 2).

Change conflicts can also arise when teams work asynchronously, making any number of changes while offline from each other. After working offline for a period of time, a user reconnecting may find that their changes are in conflict with those made by others in the group.

3.4 Stretchies

In Miramar, one of the most common forms of change conflict is in the position of objects in the scene. We provide visual feedback of these positional changes using a semi-transparent *stretch* between objects to indicate that a conflict exists (see Figure 3). A copy of the object is displayed in every possible current state with a stretched connection between every pair. Two buttons are available inside each stretch to resolve the conflict. The button closest to each state will “choose” that state.

3.4.1 Quantity and Distance

Stretchies are most effective when the number of conflicts is small and the distance(s) in the scene between the possible states is large. The number of conflicts can be large in two different ways. If many conflicts exist for a single object, there will be many copies with stretched connections of that one object. If many different objects each have a single conflict, there will be many pairs of stretched objects. In either case, the user may be overwhelmed when this number grows too large.

Another visual difficulty arises when the distance between copies of an object in conflict is small. Then the copies may overlap and the stretch between them may be small and difficult to identify. Also, the buttons available to resolve the conflict will be more difficult to acquire. To resolve the conflict, it may be necessary to increase the distance between the copies.

3.4.2 Importance of Position

Stretchies use a connection between two positions in a Miramar scene to represent a positional change conflict. This preservation of the possible positions and the evidence of conflict can hold pertinent information. Some work, such as layout plans, may depend highly on the precision of position and other work, such as brainstorming, may not place such a high importance on

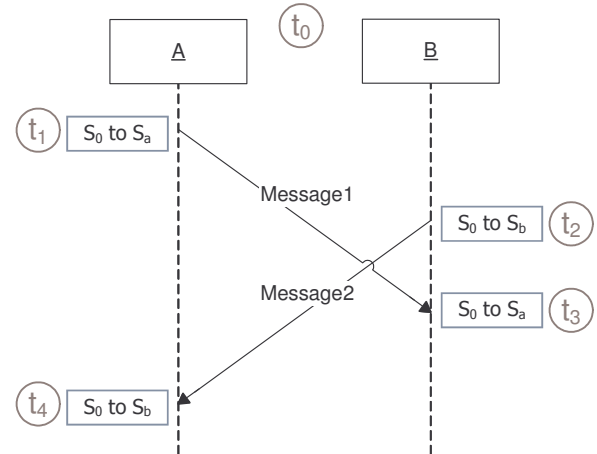


Figure 2. A message passed from client A may not be received until after client B has made a change, resulting in two possible states, S_a and S_b .

position. Some teams may also use position to communicate information to other team members. For example, objects at the front of a scene may be more important than those at the back.

When positional differences are an important part of the task, stretchies can provide much needed information about change conflicts. For work in which position does *not* play an important role, stretchies can also be useful, since they notify the user that a change conflict has occurred. Because the actual position is less important, users are also free to move objects, thus improving their ability to understand and resolve conflicts.

3.4.3 Change Information

Tam and Greenberg [17] provide a framework of information important to change awareness in synchronous and asynchronous work. According to this framework, stretchies provide information about *where* the change occurred and *what* that change was, but not about *who* made the change, *how* that change was made, nor *why* the change was made.

Presence information, such as mouse cursors and avatars are not currently available in the Miramar interface. This decision was made to encourage change conflicts to occur during the experiment. However, we do feel that this information is important to provide and is intended for inclusion in future versions of this application.

3.4.4 Capabilities

The current implementation of stretchies shows a connection between all current possible states, but the visualization is not limited to this set of states. It is also possible to represent previous states as copies of the object and to use a stretch to represent a different relationship, such as change (i.e. version) history. Users could also be provided with a control that allows them to see all changes since a certain date or time.

Although stretchies currently represent only positional changes, they are not limited to this form of change. It is also possible to represent other forms of change conflict by assigning a change to the position variable. For instance, if a team of authors is collaboratively editing a document, each copy can represent a different version. The stretching can be used to indicate either the

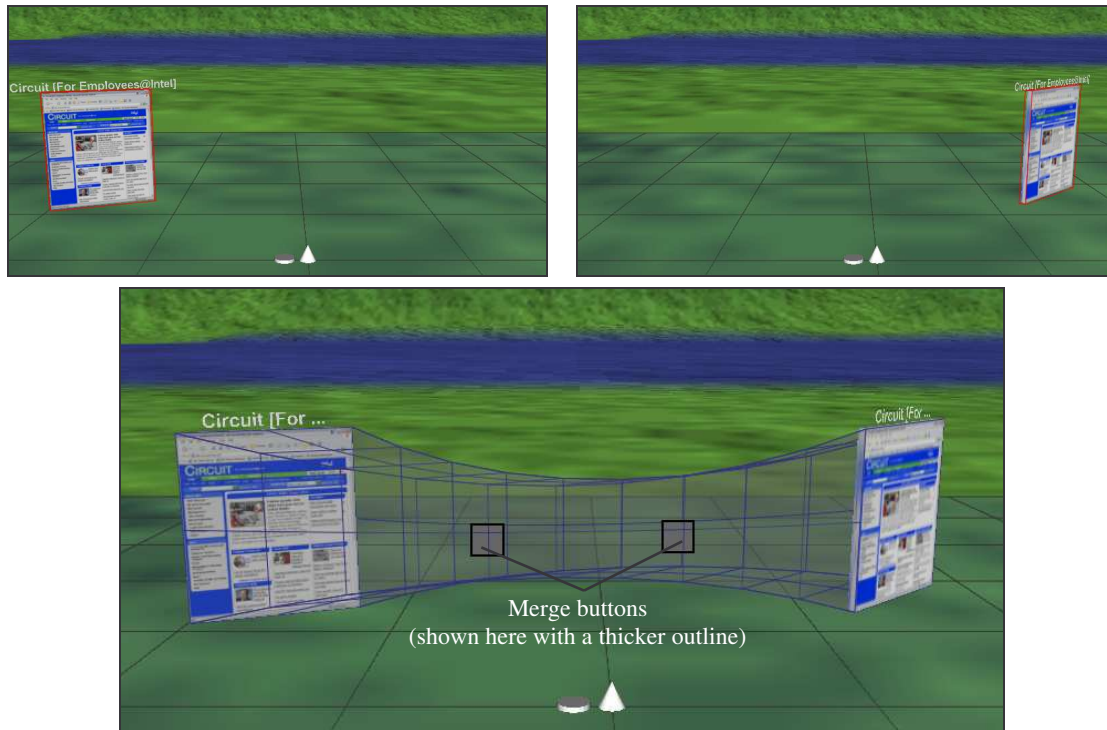


Figure 3. Visual feedback in Miramar shows the first user's changes (top-left) and the second user's changes (top-right) in the same scene as a stretched connection between both possible states (bottom).

version history or a connection between all possible current versions.

Since stretchies connect the corners of the objects in conflict, other conflicts such as changes in scale and rotation are also indicated. Since positional changes are the most common form of changes in this environment, we focus specifically on this aspect of the feedback in our discussion.

Stretchies facilitate the use of a user-controlled policy by providing information that indicates conflicts as they occur and mechanisms for resolving these conflicts. However, it is also possible for the user to choose to use a pessimistic policy, by providing a lock button on each object. Teams can choose to work according to the policy which best suits their needs and this feedback can provide them with the necessary awareness when conflicts arise as well as enabling the transition between pessimistic and optimistic policies.

4. User Study

We performed a user study to compare synchronous and asynchronous use of Miramar with and without visual feedback of change conflict (stretchies). The experiment was a 2 (synchronous vs. asynchronous) x 2 (stretchies vs. optimistic policy) mixed design.

4.1 Participants

We tested 8 synchronous pairs (4 mixed and 4 male-only) and 9 asynchronous individuals (7 males and 2 females) for a total of 25 participants. All pairs in the synchronous condition had previously met, but the asynchronous participants were strangers.

4.2 Apparatus

Two Intel® Pentium® (3+ GHz) machines each with a 19" LCD monitor were used. In the synchronous stage, one participant's computer was used both as a client and server and the other's as only a client. In the asynchronous stage, the participant's computer was used as a client and the server was run on a separate machine. In the synchronous stage, users were able to speak to one another via a speakerphone. The only visual means of communication provided was through Miramar.

4.3 Method

Before beginning the task, participants were given a brief tutorial on usage of Miramar. Participants were given instruction on navigation as well as on the meaning of the stretching visual feedback. As detailed shortly, each trial involved two stages. In the first stage, two participants *synchronously* used Miramar at two workstations in different rooms. In the second stage, a third participant used the Miramar scene *asynchronously*. Following the experiment, each participant completed a questionnaire followed by an interview.

Each user (or pair of users) performed two tasks in total: one with stretchies, and one without. Tasks were presented in counter-balanced order. When the visual feedback was absent, an optimistic policy was used to resolve conflicts, where the last update made on a given machine – either by a received message or by a user's interaction – is reflected on that machine.

4.3.1 Stage 1 (two synchronous users):

Each user was presented with the same populated Miramar scene and could communicate via speaker phone. Before beginning the first trial, both participants were given the following instructions:

“Your friend has added 10 websites which he/she thinks are potential candidates for the best <movie choice/vacation spot>. Your task, together with your teammate, is to decide on the 3 which you think are best. These can be chosen from the 10 that you see before you or from any that you choose from the web yourself. Once you finish, you must indicate which three you have chosen in whatever way you choose so that your friend can later decide on one.”

4.3.2 Stage 2 (one asynchronous user):

The third participant was later presented with the same *original* scene as the participants in Stage 1. The participant was given the following instructions:

“Imagine you are currently in your hotel room and the internet service they provide is currently not working. You previously came up with an initial set of 10 websites which you thought were potential candidates for the best <movie choice/vacation spot>. You passed these 10 websites to your friends, from which they were to choose the best 3. Your task is to review these 10 websites and organize them as you see fit. You may add any websites that you think are missing.”

The user was later connected to the changes made by the first two users; the scene was changed to reflect these two users’ updates by either showing the stretchies or by showing objects only in their new positions. The user was then told:

“The concierge at the hotel has just called your room to inform you that the internet connection has been restored. You now can go online and see what your friends have done in your ‘absence’. From the 3 tasks that they have chosen (or any that you feel may be more appropriate), determine which one you think is best. Once you finish, you must indicate which website you have chosen in whatever way you choose so that your friends can come back later and see.”

5. Results & Discussion

5.1 General observations

5.1.1 Synchronous with change awareness

Participants typically noticed conflict almost as soon as it occurred: they saw the object copy and the stretchy connections between them. In almost all cases, participants would verbally acknowledge the conflict to each other (e.g., “oh, did you see that?”), and then one person would ‘back off’ to resolve the conflict. In several cases, participants would talk about the conflict, but both would continue ‘playing’ with the split objects (we believe this was due to the novelty effect, as it typically happened early in the task and was accompanied with playful talk and laughter). Even so, one would soon release the object, again accompanied by comments such as “here you go”. Conflicts in this condition were typically resolved quickly after they appeared and so the number of conflicts present at any one time was typically small (usually one or none).

5.1.2 Synchronous without change awareness

When change conflict occurred in this case, participants would release an object over a particular location only to have it immediately move to the new position specified by their partner. Participants sometimes (but not always!) noticed this movement: one would verbally comment on it, and both would use speech to resolve conflict by agreeing who was in control. When

participants did not immediately notice the conflict, they often realized it much later, (e.g., after several minutes), by exclaiming something like “Oh, I thought I put that here”. Only then did they realize that it must have been moved by their partner.

5.1.3 Asynchronous with change awareness

A typical example of what the users saw in this part of the experiment is shown in Figure 4. When other participants’ changes were assimilated into the scene, about half the participants explored the scene before manipulating any objects. They did this to make sense of what was going on (but only one used the overview feature). The other half, however, did not explore the scene but instead started manipulating objects

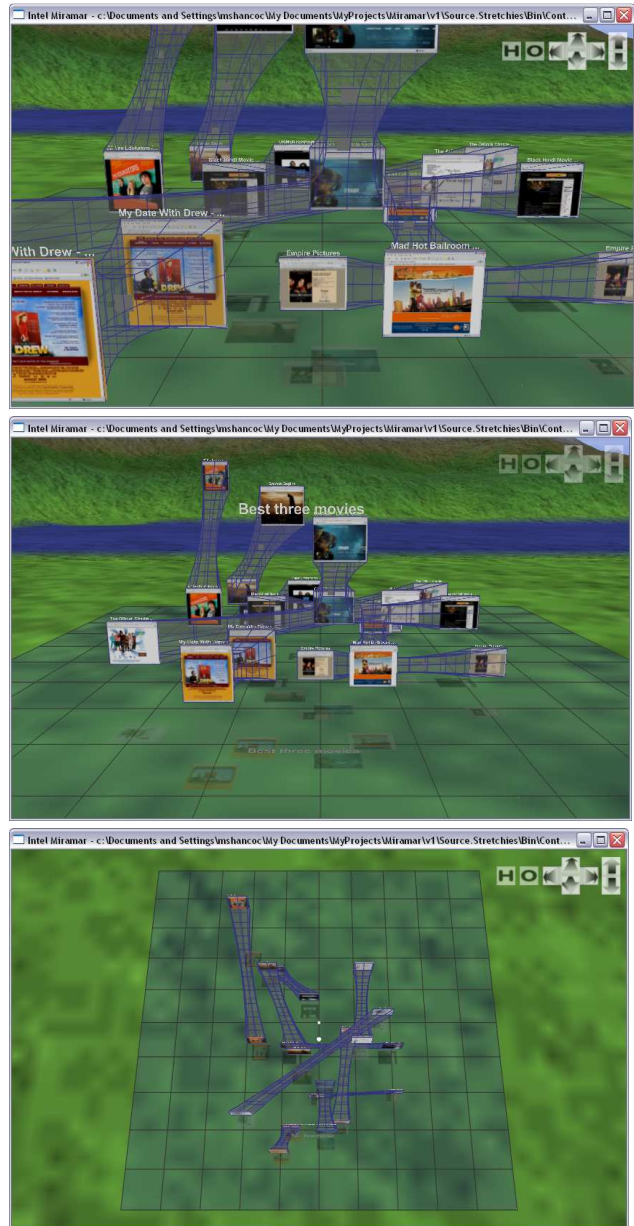


Figure 4. A typical Miramar scene in the asynchronous case when the stretching visualization was used initially (left), after zooming out (middle), and upon using the overview feature (right).

immediately. Some participants seemed somewhat overwhelmed by the many conflicts: these participants typically completed the task by leaving the conflicts in place. That is, they would move a single one of the paired objects to unique locations without collapsing them. The other participants (who did not seem overwhelmed) would typically collapse conflicting object pairs into a single preferred one. If object pairs were quite close to each other, the choice between the pair appeared arbitrary. If object pairs were far apart, the participant appeared to consider and choose the one in the best location. After collapsing was completed, the participant would reposition objects as needed.

5.1.4 Asynchronous with change awareness

When there was no visual feedback, conflicting objects always shifted to the location specified by the other people. Usually, the participant quickly inferred what objects seemed to have moved, and that these movements were likely the result of other people's actions. While they sometimes wanted to know where they had previously placed these objects, they appeared able to complete the task without further problems.

5.2 General Discussion

The results of this study show a clear difference between synchronous and asynchronous use of the stretching visualization. The most important lesson to take from these results is that these two forms of collaboration are significantly different and that, although the underlying change conflict may benefit from the same theory, the same visual feedback can have different implications to the user experience in each case.

5.2.1 Synchronous Users

The results of the study clearly show that when working synchronously, stretchies are both understandable and useful. This result was also reflected in comments made by pairs in the follow-up interviews, in which users stated a clear preference of stretching vs. the optimistic policy. One possible explanation for this preference is that, in the synchronous case, the number of concurrent conflicts is typically small, since users can resolve conflicts as they arise. As mentioned before, having a small number of conflicts is beneficial to the usability of stretchies. Also, the stretching feedback will only appear on an object that is being or has very recently been moved. From the perspective of an individual user, the visual feedback provides notification that another user is currently using the same object. Thus, the user's focus of attention is typically already on the object associated with the stretchy. Focal attention may be another cause for the increased understandability of stretchies in the synchronous case.

The success of stretchies in synchronous collaboration can be further explained by conversation theory [2,7]. In particular, stretchies support what Dix et al. [7] describe as *feedthrough*, the communication through the artifact where "one participant's manipulation of shared objects can be observed by the other participants" (p. 495). The duplication of the conflicting object gives direct feedback and feedthrough of how all participants are manipulating that object. In contrast, the situation without change awareness does not provide effective feedthrough: other person's conflicting actions are only visible after the fact and may be missed. Next, the visualization is a non-verbal form of *back-channel communication*. That is, if no conflict appears visually after one selects an object, it is a visual *acknowledgement* that the other participants see that person's actions and will not interfere

with it. If conflict does happen and appears visually, it is *negative evidence* that there is a problem. When this is seen, people then have several options. First, a producer of the conflict can immediately initiate one's own repair by backing off. As Clark and Brennan note [2]: participants "prefer to repair their own utterances rather than let their interlocutors do it" (p. 226). Second, they can move to talk to clarify and immediately repair the issue (what is normally considered repair and clarification in conversation through nested *adjacency pairs*).

In essence, supplying the stretchies (vs. not doing so) lets people efficiently collaborate in the presence of conflict: they can receive positive or negative evidence of the consequence of their actions, and they can very rapidly detect and repair breakdowns (the conflicts) as they occur.

5.2.2 Asynchronous Users

In contrast, the asynchronous participants were more often unable to understand the stretching and were far more hesitant about its use. This difference is likely due to the fact that, when a user comes online, all changes are synchronized at once. Due to the length of time and number of changes made before synchronization, the number of conflicts that appear simultaneously is far greater than in the synchronous case. This increase in conflicts can be overwhelming and may have been the cause of the heightened confusion and lack of satisfaction from the asynchronous participants.

In the asynchronous case, it is also more difficult to resolve conflicts because there is no means of real-time communication with the other users. Without the ability to discuss potential solutions or immediate feedback during communication, conflict resolution becomes more difficult. Thus, independent of any visual feedback provided, change conflicts in the asynchronous case are more confusing and difficult to resolve.

Conversation theory also explains some of the major differences seen with stretchies when comparing the asynchronous case. Even though the medium is identical (the virtual world of Miramar, the stretchy visual feedback), the asynchronous use imposes many constraints over its synchronous use that transforms how people deal with the situation. In the synchronous case, people react as if they are part of a live conversation: the Miramar environment and the stretchies inform and augment people's joint actions. In the asynchronous case, the Miramar environment and the stretchies are not by themselves sufficient to support a conversation. A person may know that something has changed (either from the stretchies or by noticing that something has moved), but does not have any of the context to ground that change. Thus it is not seen to be a continuation of a conversation, but just a new state of the environment that must be manipulated.

It has been shown previously that designing for individuals differs from designing for groups [12]. However, the distinction between asynchronous and synchronous users adds another dimension to consider. This previous literature suggests that the individual user experience may be hindered by providing change information, such as animations, since they slow down progress. In contrast, the same change feedback benefits the group user experience due to a heightened awareness of other users' actions. Although asynchronous users may be working individually, they may require *more* awareness information than synchronous users, since they do not have access to changes made in real time.

5.2.3 Improving the Asynchronous Visualization

Although the stretchies as they appeared in the experiment were clearly not as suitable for the asynchronous case as for the synchronous case, we feel that an improved design may alleviate some of the difficulties experienced. To allow users to recognize and resolve large numbers of conflicts, the stretchies could provide dynamic feedback as the user's mouse moves across the stretch. Ghosting could also be used to indicate differences between "what's yours" and "what's mine". Colouring could also be used to indicate ownership of a change made.

Future work will address these additional complications with the asynchronous case. For problems of recall, we can provide a time slider to show the sequence of changes and the time they were made. For the problem of scale, we are working on alternative visual designs that take up less space and are less prone to clutter.

6. Conclusion & Future Work

We have introduced stretchies, a form of visual feedback for change awareness in a 3D shared workspace. Our user study shows that stretchies are useful and usable for synchronous users of the Miramar environment. Our results also show that the same visual feedback may not be suitable for asynchronous users without improved change awareness information.

The main contribution of this work is to show that representations of change depend very much on how people perceive it as part of a conversation versus as indication of collective changes to a visual space. In the synchronous case, change awareness contributes to the moment-by-moment joint actions of participants. In the asynchronous case, change awareness provides a way for people to discover and repair how the scene has been altered by others. This has serious implications to how developers should design change feedback mechanisms. Ideally, we would like a single form of visual feedback to suffice for both cases. Our study shows that this is a difficult challenge.

In the future, we intend to explore improved designs for stretchies to reduce clutter and space usage and to enhance the design with other vital change awareness information. We also intend to continue studying the differences between synchronous and asynchronous use of Miramar. For instance, this study uses a between-subjects design for the synchronous vs. asynchronous factor, and so users in these two groups had unique experiences. Since teams often work both synchronously and asynchronously, it may not be feasible to use a different interface for each case. We intend to explore the support of transition between these two modes of work within the Miramar environment and to test whether or not synchronous use improves understanding and satisfaction in the asynchronous case.

7. Acknowledgements

We would like to thank members of the Collaboratory group and John Light at Intel Corporation for valuable feedback on development and design ideas in Miramar. We would like to thank the National Science and Engineering Research Council (NSERC) and Alberta Ingenuity for funding.

8. REFERENCES

- [1] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen

content on touch- and pen-operated systems. In *Proc. Interact 2003*, 57-64.

- [2] Clark, H. and Brennan, S. Grounding in communication. In R. Baecker, *Readings in Groupware and Computer Supported Cooperative Work: Assisting Human-Human Collaboration*. Morgan Kaufmann, (1992), 222-234.
- [3] Cockburn, A. Revisiting 2D vs 3D implications on spatial memory. In *Proc. Australasian User Interface*. ACM Press (2004), 25-31.
- [4] Cockburn, A., and McKenzie, B. 3D or not 3D?: Evaluating the effect of the third dimension in a document management system. In *Proc. CHI 2001*, ACM Press, 434-441.
- [5] Cockburn, A., and McKenzie, B. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In *Proc. CHI 2002*, ACM Press, 203-210.
- [6] Collomb, M., Hascoet, M., Baudisch, P., and Lee, B. Improving drag-and-drop on wall-size displays. In *Proc. GI 2005*, ACM Press, 25-32.
- [7] Dix, A., Finlay, J., Abowd, G., and Beale, R. *Human Computer Interaction*. 2nd Edition, Prentice Hall (1998).
- [8] Dourish, P. Consistency guarantees: Exploiting application semantics for consistency management in a collaboration toolkit. In *Proc. CSCW 1996*, ACM Press, 268-277.
- [9] Dourish, P. The parting of the ways: Divergence, data management and collaborative work. In *Proc. ECSCW 1995*.
- [10] Greenberg, S., and Marwood, D. Real time groupware as a distributed system: Concurrency control and its effect on the interface. In *Proc. CSCW 1994*, ACM Press, 207-217.
- [11] Gutwin, C. and Greenberg, S. A descriptive framework of workspace awareness for real-time groupware. *JCSCW*, 11, 3 (2002), 411-446.
- [12] Gutwin, C. and Greenberg, S. Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proc. CSCW 1998*, ACM Press, 207-216.
- [13] Hancock, M. S., Shen, C., Ryall, K., and Forlines, C. Exploring non-speech auditory feedback at an interactive multi-user tabletop. In *Proc. GI 2005*, ACM Press, 41-50.
- [14] Light, J., and Miller, J. D. Miramar: a 3D Workplace. In *Proc IPCC 2002*. IEEE, 271-282.
- [15] Pekkola, S. Critical approach to 3D virtual realities for group work. In *Proc NordiCHI 2002*, ACM Press, 129-138.
- [16] Sun, C., Zhang, Y., Jia, X., and Yang, Y. A generic operation transformation scheme for consistency maintenance in real-time cooperative editing systems. In *Proc GROUP 1997*, ACM Press, 425-434.
- [17] Tam, J., and Greenberg, S. A framework for asynchronous change awareness in collaborative documents and workspaces. *International Journal of Human Computer Studies*, (In Press).
- [18] Tavanti, M., and Lind, M. 2D vs 3D, implications on spatial memory. In *Proc. InfoVis 2001*. IEEE, 139-145.
- [19] Ware, C., and Franck, G. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *TOG*, 15, 2 (1996), 121-140.