Traces: Visualizing the Immediate Past to Support Group Interaction

Carl Gutwin

Department of Computer Science, University of Saskatchewan
57 Campus Drive, Saskatoon, SK, S7N 4J2
carl.gutwin@usask.ca

Abstract

Virtual embodiments of people in groupware systems provide a wealth of information to others in the group. They allow for explicit gestural communication, and they provide implicit awareness information about people's locations and activities. However, the constraints of current networked groupware limit the effectiveness of these kinds of communication. This paper investigates how embodiments can be augmented with traces – visualizations of past movements – to help others perceive and interpret bodily communication more clearly and more accurately. The paper presents a case study of traces applied to telepointers, and gives several examples of how the concept can be used to improve interaction effectiveness in groupware.

Key words: Synchronous groupware, groupware usability, awareness, interaction histories, edit wear.

1 Introduction

Shared workspace groupware allows people who are in different locations to work together in a visual task space. In these systems, people are often represented by some form of visual embodiment – a visible representation that stands in for the actual person in the computational workspace. Embodiments can take many forms, from telepointers and view rectangle in a 2D workspace [9] to fully-rendered humanoid avatars in a virtual world [2].

Groupware uses embodiments because the real bodies that they are modeled on are so useful in collaboration, as a vehicle for communication and group awareness. In real world collaborative situations, bodies provide other group members with a great deal of information: either explicitly, through gestural communication; or implicitly, by simply "giving off data" about what the person is doing [20].

However, groupware embodiments are poor approximations of their real-world counterparts. They provide only a small fraction of the information that would be provided by a real body, and display and network constraints make them harder to notice and harder to interpret than a real body.

Telepointers are a good example of this problem. Telepointers are not nearly so obvious as a real body in a physical workspace: they are small visual objects in a workspace filled with other small visual objects. Where actions or gestures by a real person in a physical space are readily noticeable, even through peripheral vision, telepointer motion often become lost in the clutter of the workspace. In addition, system factors such as processor load and network traffic further reduce the intelligibility of the telepointer, by making its motion erratic and jittery [8].

How can the usefulness of user embodiments be improved upon? One approach is to increase the expressiveness and realism of the embodiment by increasing the number of input sensors and the complexity of the rendered image. This direction is an established area of groupware research (e.g. [26]); however, these techniques are often difficult to implement using current desktop technology, and do not solve (and in fact add to) the problems caused by processor and network load.

A second approach is to augment embodiments using information that is not available in physical settings. This approach is complementary to that of making embodiments more realistic, and can address some of the specific problems that arise in the artificial world of a groupware workspace.

This paper investigates one such augmentation called traces: visualizations of an embodiment's past movements that make gestures and actions easier to see and interpret in a shared workspace. The following sections outline the foundations of the idea, and then present a case study of the application of traces to telepointer embodiments. Adding a trace to the telepointer that shows the path and motion of the pointer over the past few moments can smooth out jerky motion caused by network delays, can help people to understand what is going on when they glance at another person's work, and can allow for more concrete gestural communication. Other applications of the idea are also discussed; experience with the technique thus far suggests that it is a good example of how the "informational physics" [13] of groupware embodiments can be exploited to improve the usability of shared workspaces.

2 Motivation: How Bodies Communicate

Bodies (and embodiments) are a primary mechanism for conveying communication and awareness information: “whenever activity is visible, it becomes an essential part of the flow of information fundamental for creating and sustaining teamwork” ([20] p. 24). Bodies communicate in two main ways: through explicit gestures, and through consequential communication.

Explicit and intentional gestures are ubiquitous in face-to-face collaboration, and gesture has been often studied in CSCW research [1,24,21,25]. People use gestures to point to objects (e.g. “this one”), to indicate areas of the workspace, to demonstrate an action without really doing it, to illustrate concepts (e.g. using the gap between finger and thumb to show size), to indicate paths in the workspace, to communicate symbols (e.g. “thumbs up”), or to generate actual utterances in a language such as American Sign Language.

Bodies also convey information implicitly. Since most things that people do in a workspace are done through some bodily action, the position, posture, and movement of heads, arms, eyes, and hands provide a wealth of information to others about what’s going on. This is *consequential communication*: information transfer that emerges as a consequence of a person’s activity within an environment [20]. Although it is completely unintentional, consequential communication provides a great deal of information. Norman [17] provides an example from commercial aviation:

> When the captain reaches across the cockpit over to the first officer’s side and lowers the landing-gear lever, the motion is obvious: the first officer can see it even without paying conscious attention. The motion not only controls the landing gear, but just as important, it acts as a natural communication between the two pilots, letting both know the action has been done. ([17] p. 142)

Gestures and consequential communication have certain requirements in order to be seen and interpreted correctly. First, the actions must be noticeable: the large motions of arms and hands draw attention to the fact that something is happening, attention that is needed before communication can take place. Second, the information contained in gestures and consequential communication is based on motion over time, and so has temporal requirements for intelligibility. That is, a gesture must be distinguishable from events before and after it, and must be shown smoothly without too many stops and starts, in order to be successfully interpreted by another person.

These requirements are well met in the real world, where the constraints of body mechanics ensure that motions are large, and where visual information flow is instantaneous and smooth. However, in the artificial world of a groupware workspace, where embodiments are small and where system factors get in the way of information flow, these necessities are rarely provided.

3 Solution Approach: Interaction Histories and Informational Physics

Our approach to solving these problems is based on Hill and colleagues’ work on information physics and interaction histories [13,14]. Their overall motivation is the question of how computation can be used to improve “the reflective conversation with work materials” ([13] p. 3). Their research in this area arises from an important distinction between the real world and a virtual one – that each has a set of rules and physical laws that govern how people perceive information and interact with objects. The real world’s laws are not negotiable, but the “informational physics” of an artificial world are completely up to the designer. Although there are advantages to duplicating the real world’s laws – namely that people already have extensive knowledge of how they work – there are situations where changing the rules can be beneficial for the user:

> These same techniques also allow us to create virtual worlds that give concrete existence to abstract entities operating according to a physics of our choice. The entities and their physics can be designed to highlight aspects of phenomena not normally available to us but that are important for supporting understanding and task performance.” ([13], p. 7)

Thus, information in a virtual world can behave in ways that are appropriate to its meaning and importance, rather than ways determined by its physical properties. Hill et al. propose an alternate informational physics for work artifacts that bends time and shows the past in the present:

> The basic idea is to maintain and exploit object-centered interaction histories: record on computational objects…the events that comprise their use…and display useful graphical abstractions of the accrued histories as part of the objects themselves.” ([13], p. 3)

This idea leads to a number of innovative displays, such as a scrollbar that shows how often each line of a file has been read or edited. Others have followed this lead and proposed other techniques and displays [6,15,23]. However, researchers in this area have focused on work artifacts and the interactions that people have with them, and do not consider the objects in the workspace that represent people – i.e. embodiments – and the interactions that happen between people in collaboration. The intention of our research is to extend the idea of informational physics and interaction histories to embodiments, in order to make gestures and consequential communication more understandable.

The problems that we are trying to solve are ones of motion over time, and real-world physics do not hold well enough in groupware to meet their requirements. However, it may not be necessary to invent a completely new information physics for embodiments, for several alternates already exist in the worlds of the visual arts. In particular, there are art forms where movement is regularly augmented in order to assist comprehension: the cartoons and the comics.

4 Inspiration: Motion Lines in the Comics

Cartoon and comic artists have long had to address the problem of showing movement convincingly and comprehensibly in a static medium. The depiction of motion in a single image has a long history (summarized in [16]) beginning with Duchamp's and Marey's experiments with overlaid representations and with lines to trace the path of the moving object. These ideas were taken up by comic illustrators, and evolved into three distinct techniques for showing motion: motion lines, motion blur, and stutter blur. These are shown in Figure 1. Motion lines are the simplest, with one or more lines tracing the path of the moving object; motion blur adds the optical effect of streaks along the path; and stutter blur shows several intermediate representations of the object along the path.

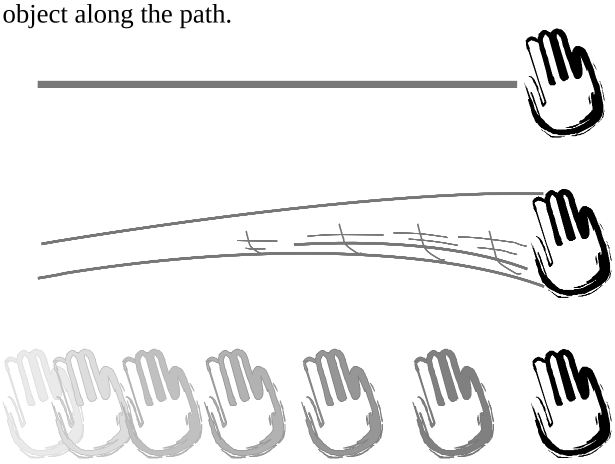


Figure 1. Motion lines, motion blur, and stutter blur in comic strip art (adapted from [16]).

Cartoonists and animators also have to deal with the problem of motion depiction. Even though cartoons depend partially on the illusion of motion created by the sequence of still frames, they generally have a much lower frame rate than film, and so must find ways to make movement seem smooth and understandable. Cartoonists regularly use both motion blur and stutter blur—not necessarily to show the path of a moving object, but more to emphasize certain movements and to make objects and characters more convincing and real to the viewer. Chang and Ungar [3] state a rule of thumb that if an object moves more than half its own size between two frames, motion blur must be used to convey the illusion of continuous movement.

In emphasizing certain aspects of a character's motion, the cartoonist provides visual cues to assist comprehension. As Chang and Ungar state, the techniques work extremely well, allowing even impossible motions and events to be easily understood. In groupware, we also want to assist understanding of others' motion and activity. The techniques used in comics and cartoons for depicting and emphasizing motion appear to be a useful alternative physics for embodiments, a way of augmenting the basic representation to better convey motion-based information. The next sections describe a case study of applying the idea of traces and the visualization techniques described above to groupware telepointers.

5 Designing Telepointer Traces

Telepointer traces are visualizations of the previous motion and location of a remote mouse cursor. Our goals in adding traces to telepointers are to make gestures easier to see, to make motion easier to interpret, and to provide a bit of context that helps people understand what is going on when they look at the telepointer. These goals, however, cannot be met at the expense of people's ability to carry out their individual work; in particular, telepointer trails cannot be distracting or annoying to the people in the workspace.

We designed several representations for telepointer trails and demonstrated them to users who had experience with synchronous groupware. The different representations used different values for the following variables:

- technique: motion lines, motion blur, stutter blur, or a combination of techniques
- trace length: the amount of motion that the trace will capture and display
- trace area: the number of pixels required for the total visualization at any one time (e.g. width of the motion lines, width of the blur, size of the stutter images)
- contrast: how clearly the trace stands out from the background
- fading: whether old sections of the trace fade out and disappear

Five examples are shown in Figure 2. Motion lines were implemented simply by joining up each of the points that the telepointer passed through, and motion blur by using a rectangle the same height as the telepointer (essentially a thick motion line). Stutter blur was implemented by drawing a copy of the telepointer at each point. Fading was implemented by making the older sections of the trail more and more transparent until they disappeared from view.

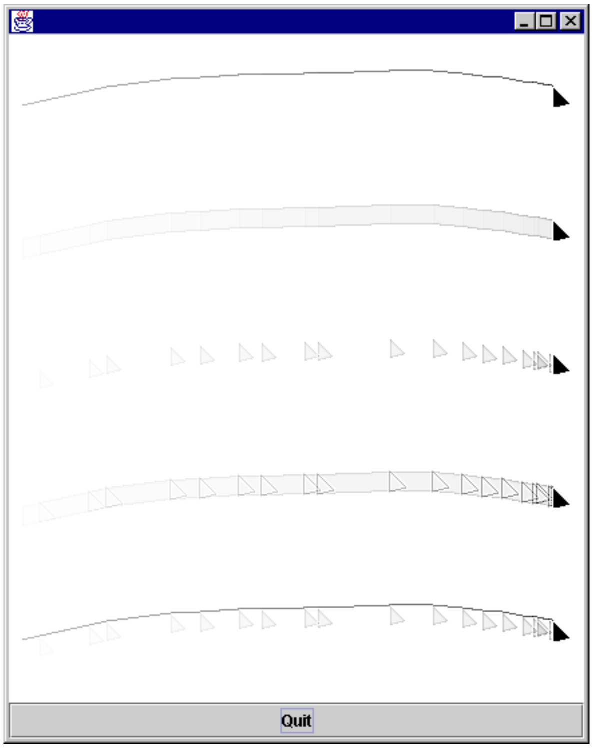


Figure 2. Example telepointer-trace representations From top: motion line, motion blur, stutter blur, motion blur plus stutter, motion line plus stutter. The actual telepointers are at right, coloured black.

Although we were more interested in simply exploring the representation space than we were in choosing the best representation, there was a general preference for short, low-contrast, fading trails that used either the motion line or motion blur technique. Some of our specific findings were:

- Traces that captured more than about one second of motion quickly cluttered the screen with lines (the "scribble effect"), reducing the clarity of the more recent lines. Many gestures and movements take less than a second to produce, and there seems to be little value in showing more than the latest one.
- Low contrast traces were seen as less distracting than high-contrast trails. However, low-contrast traces were still surprisingly easy to see, even when the darkest part of the trail was 90% transparent.
- Fading trails seem much more natural than solid trails (those of uniform colour for the entire length of the trace). Since trails are added to at one end and removed at the other, solid trails appear to be distinct objects that moves as a unit. This gives the odd impression of a worm chasing the telepointer.
- Stutter blur on its own does not do a good job of showing the path traveled. Especially when the telepointer is moving quickly, stutter blur leaves gaps that can be difficult to interpret. However, stutter blur does record changes in pointer speed, which lines or blur alone do not.
- Many of the representations were computationally intensive, a distinct disadvantage in something intended to offset the effects of processor load. Single lines were predictably the cheapest to draw, and the combination representations (e.g. motion blur plus stutter) were most expensive. However, transparency appeared to be the biggest drain on the CPU, regardless of which technique it was used with.

6 Example Applications of Telepointer Traces

We have added telepointer traces to a variety of simple groupware systems in order to explore the concept further and to determine whether traces assist group interaction. Two of these investigations are described below: using traces to combat jitter, and using traces in realistic groupware applications.

6.1 Using traces to smooth network jitter

Network jitter is the intermittent delay due to variance in the arrival times of a stream of messages (such as telepointer position messages). These intermittent delays cause the telepointer to momentarily freeze on the screen, and also cause messages to pile up, with several arriving at the receiver at once instead of being correctly spread out. When the receiver processes the multiple messages, several telepointer moves are collapsed into a single screen update. The telepointer looks as if it is jumping from position to position, and the end result is a frame rate so low that the illusion of smooth motion is impossible to maintain [8].

We implemented a telepointer trace using a motion line to smooth out this jittery movement. Since all of the telepointer positions are received (but not all are shown), it is possible to recreate the actual motion of the pointer using the trace. Thus, a viewer should get a more complete representation of the gesture than without the trace. An example of the effect is shown in Figure 3.

We are currently comparing people's abilities to recognize basic pre-recorded gestures (such as the outlines of number and letter shapes, paths, and enclosure of areas) at various levels of network jitter, both with and without the telepointer trail. The trail was designed so that about one-third of the gesture was visible in the trail at any one time. Our initial results show that at high jitter magnitudes, the trail makes an enormous difference in interpreting gestures. The reason is simple: without the trail, the viewer sees only a few frames of the gesture – that is, the telepointer jumps to only a few points and no smooth motion is apparent at all. With the trail, all of the intermediate points that were skipped are drawn as trail segments, and the viewer sees all of the movement (although not smoothly). The gesture is essentially drawn on the screen, and the viewer is simply to look at it, rather than attempt to interpolate between a few isolated frames. More details on this study can be found in [7].

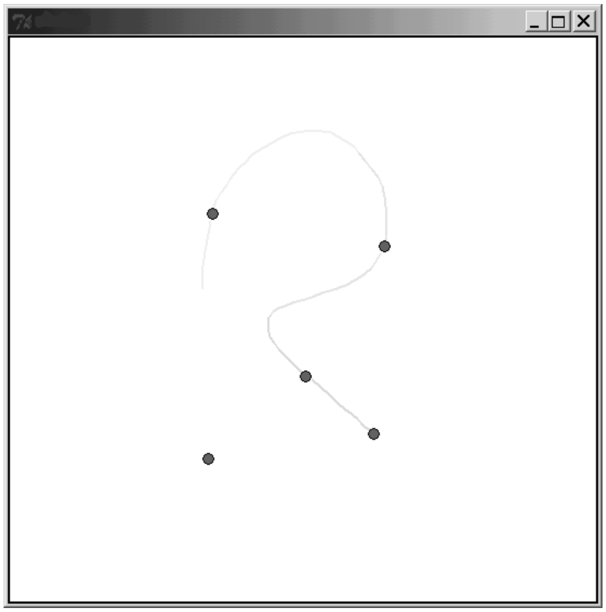


Figure 3. Telepointer gesture with trail. The dots indicate the only telepointer positions that the viewer would see in a high-jitter situation.

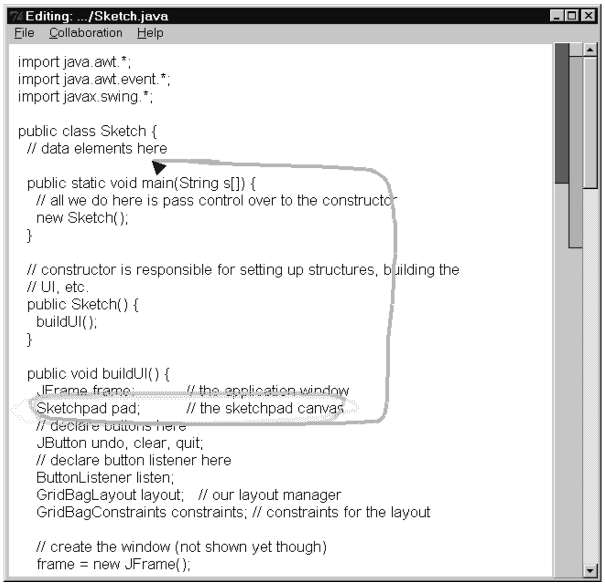
6.2 Trails in realistic groupware applications

We implemented telepointer trails in a group file viewer and in a real-time multiplayer game, both of which are Groupkit applications [18]. In these examples, we were interested in finding out how the traces were used in a more naturalistic setting. We have tested the applications informally with several groups, both with people in our lab and with outsiders.

We again used the motion line representation to reduce computation load. In addition, Tcl/Tk (the language underlying Groupkit) does not allow transparency, so we implemented fading by gradually changing the line colour to match the background, and applying a bit-mask to gradually remove the line.

Figure 4 illustrates the shared file viewer. People are represented in the document workspace with telepointers (the system also includes a multi-user scrollbar to show out-of-view location). This system is intended for use as a discussion tool (such as in a code review); therefore, people will generally have a shared focus on the document. In the figure, one person is indicating a variable declaration and where in the file it should be moved.

Second, Figure 5 shows a multi-player game similar to the Pipedreams arcade game. The object of the game is to connect an inlet valve to an outlet by dragging pipe pieces from storehouses and attaching them to the end of the pipeline. The players must work quickly, since after a certain amount of time water begins flowing through the pipe, and will flood the workspace if it reaches the open end of the pipe. This system provides a large workspace and gives more opportunity for independent work in different parts of the workspace. In the figure, one participant has just dragged a pipe from the stack at lower left to the end of the partially completed pipeline, and has moved back towards the left side of the screen.



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Sketch {
  // data elements here

  public static void main(String s[]) {
    // all we do here is pass control over to the constructor
    new Sketch();
  }

  // constructor is responsible for setting up structures, building the
  // UI, etc.
  public Sketch() {
    buildUI();
  }

  public void buildUI() {
    JFrame frame;          // the application window
    Sketchpad pad;         // the sketchpad canvas
    // declare buttons here
    JButton undo, clear, quit;
    // declare button listener here
    ButtonListener listen;
    GridBagLayout layout;   // our layout manager
    GridBagConstraints constraints; // constraints for the layout

    // create the window (not shown yet though)
    frame = new JFrame();
```

Figure 4. Telepointer traces in a code browser.

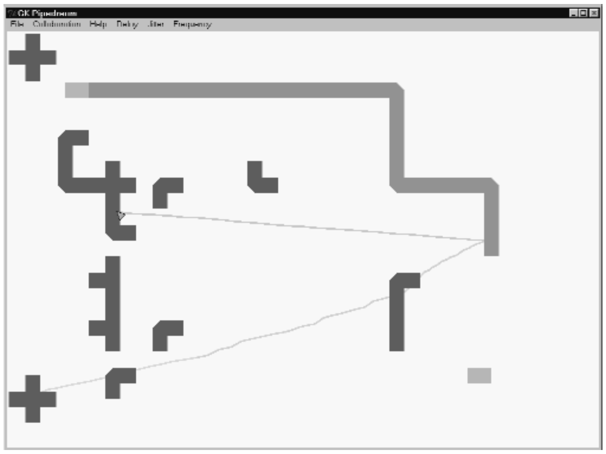


Figure 5. Telepointer traces in a multi-player game.

7 User Experiences

We discussed the traces with participants after they had used the groupware systems both with and without the telepointer traces. Our evaluations were unstructured and opportunistic; our goal was to see whether traces would be useful for helping people maintain awareness and for assisting them in communicating with gestures.

People in general liked the idea of telepointer traces, and had no difficulty understanding what they were for. Furthermore, none of the participants found the traces to be particularly distracting in any of the applications. In the multi-user game (where people have to concentrate on their individual tasks), players said that it was quite easy to ignore their partner's trail when they were concentrating on their own work. A few people suggested, however, that putting the representation under the control of the viewer would allow people to reduce distraction when it did occur.

7.1 Using Traces for Awareness

Our intention was that telepointer traces would assist awareness of where people were and what they were doing. In our discussions with people after they used the systems, some participants said that they had a better sense of what activities the other person was carrying out. For example, one person used the line of the trail to predict where the other person was moving in the pipedreams game. A second person said that they used the traces as a general indicator of activity – that is, he could see "out of the corner of his eye" if the other group members were active

However, people did not often recall situations where they gathered awareness information by watching the traces. On reflection, it seems that for many tasks, the basic telepointer alone provides enough consequential communication to support the collaboration adequately. The realistic groupware systems were run on a fast network, however, and we believe that there are certainly situations where the normal telepointer will not be able to convey enough information for adequate awareness. In addition, certain types of tasks have greater requirements for keeping track of people's precise actions. For example, design work or instructional situations often require keeping a closer eye on others' activities. Finally, one participant suggested that if the application itself involved gestural input (e.g. [12]), the trails would be extremely helpful in understanding what another person was doing.

7.2 Using Traces for Gesturing

In contrast, almost all of the participants used the traces for gestural communication. People found that gestures were much easier to see, and people liked the way that their gestures were made more persistent by the telepointer trace. Several participants said in particular that indicating objects in the workspace was easier with the traces. For example, people would circle lines of code in the file browser, or would draw out a path for the pipeline in the multi-user game. We were initially surprised at the number of positive comments about *creating* the gestures (instead of viewing them). However, communicative acts require both that the receiver interprets the message successfully, *and* that the sender gets confirmation that the message has been received. We believe that telepointer traces assist this confirmation process, in that the creator of the gesture had greater confidence that the receiver was actually seeing the gestures correctly.

The "persistent gestures" that are afforded by telepointer traces blur the line between gesturing and drawing. Many kinds of gestures are really "drawings in the air" in the first place, and adding a trace to the pointer simply builds on the idea. Drawing makes it easy to see an entire gesture at once and allows relationships to be shown over a larger distance, while the gradual fading prevents the workspace from becoming cluttered with marks. Using traces for drawing is similar to a groupware feature that has appeared in several systems, that of providing an annotation layer for drawing on the workspace without changing it [19]. We consider traces as a very lightweight version of this annotation layer, and traces could easily be extended to produce permanent marks as well. However, it seems clear that the transitory nature of traces has a place in collaborative communication. People liked being able to mark up the workspace in an impermanent way, although they wanted more control over how long the traces stayed around.

Overall, people felt that traces improved gestural communication for both the creator and the viewer of the gesture. Since mice are imprecise and telepointers are small, the lines drawn by the telepointer trace helped people to indicate objects, show paths, and draw associations.

8 Extending the Idea Beyond Telepointers

Telepointers are only one type of embodiment, and there are other possibilities for interaction histories with other types. In addition, there are other kinds of informational physics that could be used with other representations. Two examples are described below.

8.1 Viewport rectangles

A second common type of embodiment in 2D shared workspaces is the viewport. This is simply a rectangle superimposed on the workspace indicating the bounds of each person's view, and is useful for knowing when an object can be seen by another person. View rectangles are often implemented in miniature overviews of the entire workspace (called radar views [22]).

Interaction histories can also be maintained for viewports; although the viewport is not involved in gestural communication, it is often useful for group members to know where another person has been working in the workspace. An example radar view where viewports are augmented with traces is shown in Figure 6. As with the telepointer trace, a fading trail shows the past locations and movements of the viewport. The informational physics of viewports are slightly different that those of telepointers, however: the time scale of interest for a viewport is much larger, since they change position much more slowly than telepointers. Therefore, a trace would show the previous minutes or hours of viewport movement, rather than only the past few moments. Viewport traces are another example where allowing user to control the time period shown in the trace could be valuable. For example, a person might wish to extend the coverage of the visualization backwards in time to see whether a person had ever visited a particular region of the workspace.

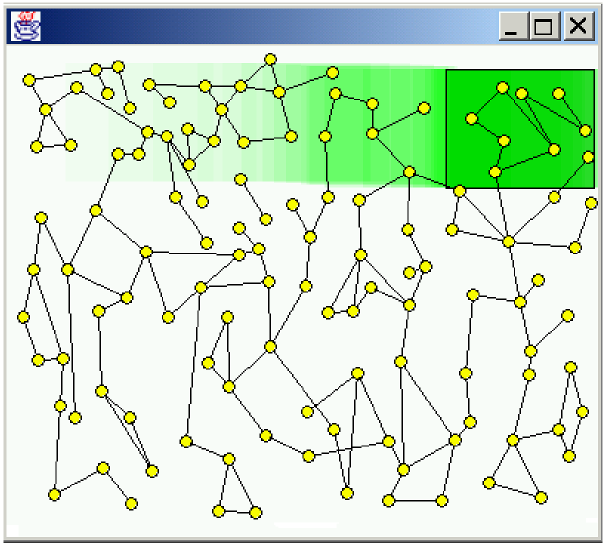


Figure 6. Radar view on a graph browser, with fading trail on viewport (colours darkened for printing). Main view is not shown.

8.2 Casual awareness with video

In video-based collaboration systems, people are represented by video images of themselves, rather than by a graphical embodiment. Nevertheless, we can apply the idea of traces to add information about past activities. For example, consider an awareness server that lets a workgroup see regularly-updated video snapshots of each others' offices, to help maintain casual awareness of who is around and who is available for collaboration (e.g. [4]). Since people only look at the pictures sporadically, they may miss useful events, such as someone returning from lunch but then leaving their office again.

Adding a trace to each person's video image provides information about activities and presence over a particular time period. For example, Figure 7 shows how a system could composite the last five snapshots from a webcam to provide awareness of the past five minutes in a person's office. Previous images appear as ghosts overtop the current snapshot; in the example situation given above, a co-worker would now be able to tell that the person had at least been in the office, even though they were no longer there.

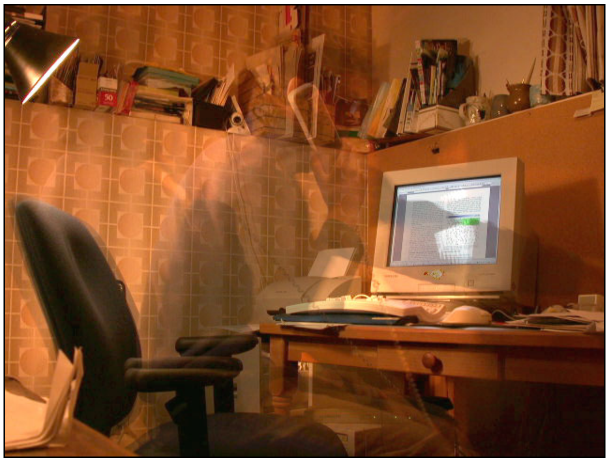


Figure 7. Composite image with stutter blur, achieved by overlaying multiple images. Current image is of the empty office; previous images appear as ghosts.

9 Conclusions

CSCW research usually attempts to support the interaction techniques and mechanisms that people already know from face-to-face collaboration. This approach is a necessary one, but the traces project suggests that there are also situations where computation can be used to augment reality and help overcome the drawbacks and limitations of distributed groupware environments.

This paper has shown examples of how embodiments can be augmented using an informational physics that visualizes the past along with the present. The past is visualized as traces of movement and activity. When applied to telepointers, traces can have significant impact on people's ability to see and interpret gestures, and can provide people with a new means of marking and annotating the workspace in a clear but temporary fashion.

Future work will include further explorations of how traces support consequential communication in longerterm tasks, more focused studies of traces in situations of high network latency and jitter, and continued development of other types of traces such as the viewport and video prototypes discussed above. In addition, there are interesting usage issues to be explored further, such as how to protect against misuse of the information summarized in a trace, how to easily control the length and duration of a trail, and how the viewer and the producer can negotiate a length and duration that are acceptable to both parties.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] 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 CSCW. *Proceedings of DIS'95 Symposium on Designing Interactive Systems*, 1995, 157-166.

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

[3] Chang, B., and Ungar, D., *Animation: From Cartoons to the User Interface*. Sun Microsystems Technical Report TR-95-33, March 1995.

[4] Dourish, P., and Bly, S., Portholes: Supporting Awareness in a Distributed Work Group, *Proc. ACM CHI'92*, Monterey CA, 1992, 541-547.

[5] Dourish, P., and Bellotti, V. Awareness and Coordination in Shared Workspaces, *Proc. ACM CSCW'92*, Toronto, 1992, 107-114.

[6] Eick, S.G., J.L. Steffen, and E.E. Sumner, SeeSoft: A Tool for Visualizing Software, *IEEE Trans. on Software Engineering*, Vol. 18, No. 11, Nov. 1992, 957-968.

[7] Gutwin, C. and Penner, R. (in preparation). The Effects of Telepointer Trails on Gesture Recognition in Jittery Groupware. In preparation, 2001.

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

[9] Gutwin, C., and Greenberg, S. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. In press, *Journal of CSCW*.

[10] Gutwin, C. and Greenberg, S. Design for Individuals, Design for Groups: Tradeoffs between power and workspace awareness. *Proceedings of ACM CSCW'98*, Seattle, ACM Press,1998, 207-216.

[11] Gutwin, C., and Greenberg, S. Effects of Awareness Support on Groupware Usability. *Proceedings of ACM CHI'98*, Los Angeles, 1998, 511-518.

[12] Hardock, G., Kurtenbach, G., and Buxton, W. A Marking Based Interface for Collaborative Writing. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1993, 259-266

[13] Hill, W.C. & Hollan, J.D. Edit wear and read wear. *Proc. ACM CHI'92*, 1992, ACM Press, 3-9.

[14] Hill, W.C. & Hollan, J.D. History-enriched digital objects: prototypes and policy issues. *The Information Society*, 10(2), 1994, 139-145.

[15] Ishii, H. "The Last Farewell": Traces of Physical Presence, *interactions*, 1998, 5 (4), 56-57.

[16] McCloud, S. *Understanding Comics*. Kitchen Sink Press, 1994.

[17] Norman, D., *Things That Make Us Smart*, Addison-Wesley, Reading, Mass., 1993.

[18] Roseman, M. and Greenberg, S. Building Real Time Groupware with GroupKit, A Groupware Toolkit. *Transactions on Computer-Human Interaction*, 1996, 3(1), 66-106.

[19] Roseman, M. and Greenberg, S. GroupKit: A groupware toolkit for building real-time conferencing applications. In *Proc ACM CSCW'92*, 43-50.

[20] 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.

[21] Short, J., Williams, E., and Christie, B., Communication Modes and Task Performance, in *Readings in Groupware and Computer Supported Cooperative Work*, R. M. Baecker ed., Morgan Kaufmann, Mountain View CA, 1976, 169-176.

[22] Smith, R. What You See Is What I Think You See, *SIGCUE Outlook*, 21(3), 1992, 18-23.

[23] Tam, J., McCaffrey, L, Maurer, F. and Greenberg, S. *Change Awareness in Software Engineering Using Two Dimensional Graphical Design and Development Tools*. Report 2000-670-22, Department of Computer Science, University of Calgary.

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

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

[26] Weimer, D., and Ganapathy, S. K. A Synthetic Visual Environment with Hand Gesturing and Voice Input. *Proc ACM CHI'89*, 1989, 235-240.