Computers & Graphics 33 (2009) 130–138

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Using mobile group dynamics and virtual time to improve teamwork in large-scale collaborative virtual environments

Trevor J. Dodds *, Roy A. Ruddle

Visualization and Virtual Reality Research Group, School of Computing, University of Leeds, UK

ARTICLE INFO

Article history:
Received 1 September 2008
Accepted 13 January 2009

Keywords:
Collaborative virtual environments
Virtual reality
Asynchronous collaboration
Group dynamics

ABSTRACT

Mobile group dynamics (MGDs) assist synchronous working in collaborative virtual environments (CVEs), and virtual time (VT) extends the benefits to asynchronous working. The present paper describes the implementation of MGDs (teleporting, awareness and multiple views) and VT (the utterances of 23 previous users were embedded in a CVE as conversation tags), and their evaluation using an urban planning task. Compared with previous research using the same scenario, the new MGD techniques produced substantial increases in the amount that, and distance over which, participants communicated. With VT participants chose to listen to a quarter of the conversations of their predecessors while performing the task. The embedded VT conversations led to a reduction in the rate at which participants traveled around, but an increase in live communication that took place. Taken together, the studies show how CVE interfaces can be improved for synchronous and asynchronous collaborations, and highlight possibilities for future research.

1. Introduction

Collaborative applications in general may be classified in terms of time (synchronous vs. asynchronous) and space (co-located vs. remote) [1]. For example, applications using shared tables and shared wall displays provide for co-located and *synchronous* interaction. Leaving post-it notes in a shared space (or using software which provides the digital equivalent on a single shared display) is an example of co-located and *asynchronous* interaction. Collaborative virtual environments (CVEs) are one way of enabling remote collaboration. They allow virtual co-location of people who are physically remote, by providing a 3D virtual spatial world for people to co-exist in.

Historically, users have had difficulty in understanding the actions of others in CVEs [2,3], and the problems mushroom in a large-scale environment (e.g., a virtual building or city) because of the extra challenges of navigating and locating the whereabouts of one's collaborators. To help with this we have developed techniques called mobile group dynamics (MGDs), which helped groups of people work together while they traveled around large-scale CVEs [4].

This paper: (a) addresses shortcomings in MGDs, which centered on the time it took users to regroup in a place to discuss or see what each other was interested in, and (b) implements the concept we call virtual time (VT) that allows (to a certain extent) virtual synchronization of people who are physically separated in time. Taken together, our techniques allow both synchronous and asynchronous collaborations in large-scale CVEs. The following sections describe the background and implementation of both suites of techniques (our updated version of MGDs and VT), and then experiments evaluate both. The MGDs work was previously reported in [5], but the VT research is entirely new. Our hypothesis was that the teleporting, awareness and multiple views functionality would improve teamwork. To analyze teamwork, we looked for improvements in two specific areas. First, we wanted to tackle problems of participants spending time collocating to communicate (or waiting until they are collocated before they talk to each other). Second, we wanted to help people work as a team by providing an awareness of the actions and perspectives of others (multiple views tackling problems 1 and 2). These were analyzed using the quantitative data provided by the server's log of activity, and a conversation transcript.

2. Methods for real-time collaboration

Previous research showed how even a basic set of MGDs techniques helped users communicate while they traveled around a virtual urban development and reviewed its design [4]. However, two major areas for improvement were also identified. First, participants tended to spatially regroup to discuss their findings, even though MGDs allowed communication over an infinite distance (there was no distance attenuation for audio

* Corresponding author. Tel.: +447752239057.
E-mail addresses: trev@comp.leeds.ac.uk (T.J. Dodds), royr@comp.leeds.ac.uk (R.A. Ruddle).

0097-8493/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2009.01.001

communication between group members). This meant that unnecessary amounts of time were spent traveling to meeting places. Second, if participants wanted to see what others were looking at (e.g., a point of interest that was being discussed) then they had to 'walk' to the appropriate location.

These shortcomings in real-time (i.e., synchronous) collaboration were tackled by adding new functionality to MGDs, taking advantage of the fact that CVEs do not need to be bound by real world constraints [6]. This new functionality: (1) used visual feedback to provide 'awareness' about who was receiving audio at a given moment in time and who was speaking, (2) supplemented a participant's own (main) view by small viewports that showed the views of fellow group members, and (3) implemented teleporting so participants could move directly to any point in the environment by clicking on it ('walking' is time consuming).

The basic MGDs techniques incorporated an explicit hierarchical grouping system, represented using a 'group graph' metaphor, and methods to assist movement as a group. The awareness functionality (see Fig. 1) used a head-up display (HUD) to display the faces of all participants who were within hearing range of you at a given moment in time (this included all participants in one's own group because there was no distance attenuation for intra-group audio communication). These faces were photographs of the participants (extracted from their photographic avatars), so they could be easily recognized. This was designed to make the participant aware that they could be heard by all the participants shown on their HUD, even if some of them were fellow group members whose avatars were a considerable distance away. When another person was talking, their face was highlighted on the HUD, with a speech icon next to it. This gave participants additional information as to who was speaking, which was particularly useful if the associated avatar was out of sight.

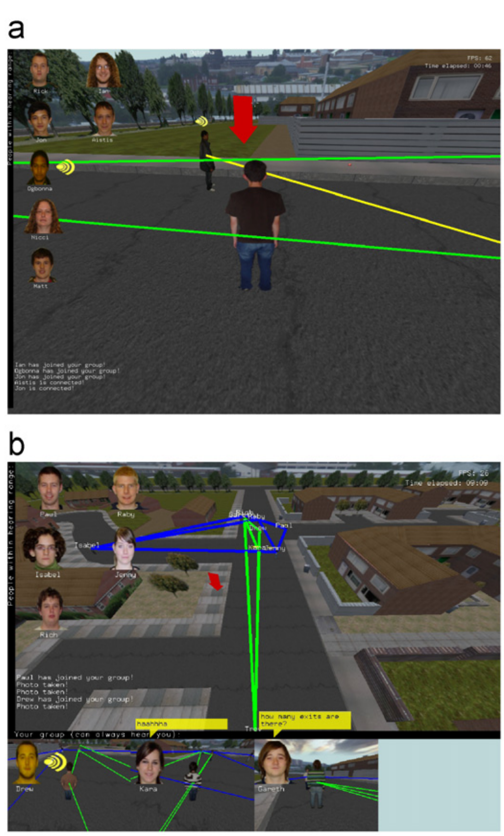

Fig. 1. Screenshots of the environment in the two conditions: teleport and multiple views. The graph metaphor, speech icon, teleporting arrow, and participants within hearing range can be seen in both figures. The views of fellow group members can be seen along the bottom of the screen in (b). (a) Teleport condition, shown using an over-the-shoulder view. (b) Multiple views condition, shown using a bird's-eye view.

In VEs, users experience two kinds of problems understanding the actions of others. (1) 'Fragmented views', where another participant refers to an object or point of interest in the environment, but their avatar and the point of interest are not simultaneously visible in the viewport [2]. (2) What you see is *not* what I see, which makes it difficult to understand another's perspective. A combination of these two problems occurs if two users wish to meet at a point of interest. This is a 'Come here! Look at this' scenario (see [7, p. 136]), where the respondent needs to know the location of the user who is talking (they are unlikely to be within the viewport, see problem 1), and what they are referring to (problem 2).

To overcome these problems, Wössner et al. [8] provided a 'what you see is what I see' (WYSIWIS) view in their CVE, which would eradicate problem 2. They designed two CVE interfaces, one of which provided a master/slave style view (where one participant had complete control), and the other which provided a more flexible approach where participants still had some independence (they could change orientation). However, it was found that users preferred the independent viewpoint, so they did not interfere with the other participant. Sonnenwald et al. [9] found that users saw a benefit in both independent views and shared perspectives—users liked to be able to figure things out on their own and then discuss them collaboratively. Therefore, we provided each participant with a main window (their own view of the world) and thumbnails showing the view of each of their fellow group members (see Fig. 1).

The teleporting was implemented as rapid but visually continuous movement, rather than a sudden 'jump' to the new location. This was to help prevent disorientation associated with an instantaneous change of location [10]. The teleporting algorithm took its inspiration from [11], with the addition of gradual acceleration as well as deceleration, and to avoid problems caused by traveling through walls and hedges, raised a participant to a birds-eye view so they could clearly see where they were being taken. Teleporting was achieved either by clicking on a particular place in the VE scene, or on a fellow group member's thumbnail view (this teleported you to be next to that person). Our hypothesis was that the teleporting, awareness and multiple views functionality would improve teamwork. To analyze teamwork, we looked for improvements in two specific areas. First, we wanted to tackle problems of participants spending time collocating to communicate (or waiting until they are collocated before they talk to each other). Second, we wanted to help people work as a team by providing an awareness of the actions and perspectives of others (multiple views tackling problems 1 and 2). These were analyzed using the quantitative data provided by the server's log of activity, and a conversation transcript.

3. Methods for VT collaboration

Traditional CVEs bring together people who are physically remote, and adding VT makes it easier for people to collaborate even if they are not in the CVE at the same time. In other words, combining VT with a CVE allows asynchronous, remote collaboration. There are few examples of VT being implemented in CVEs, but exceptions are 'temporal links' to playback recorded content (e.g., 3D flashbacks to tell a story), which in some cases was activated by a production crew working behind the scenes [12], and in a second example the links were represented as virtual objects that a user could interact with to playback a recording or send messages to other users [13].