Collaborative Interaction in Virtual Environments

by

Trevor James Dodds

Submitted in accordance with the requirements for the degree of Doctor of Philosophy.

The University of Leeds
School of Computing

March 2009

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated overleaf. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Declarations

Some parts of the work presented in this thesis have been published in the following articles, which are reproduced in the appendix. All the material in these papers is the candidate's own work under the supervision of Dr. Roy Ruddle.

T. J. Dodds and R. A. Ruddle. Mobile group dynamics in large-scale collaborative virtual environments. In *Proceedings of IEEE Virtual Reality (VR '08)*, pages 59–66. IEEE Press, 2008.

T. J. Dodds and R. A. Ruddle. Using teleporting, awareness and multiple views to improve teamwork in collaborative virtual environments. In *Proceedings of the 14th Eurographics Symposium on Virtual Environments (EGVE '08)*, pages 81–88. The Eurographics Association, 2008.

T. J. Dodds and R. A. Ruddle. Using Mobile Group Dynamics and Virtual Time to improve teamwork in large-scale Collaborative Virtual Environments. *Computers and Graphics*, 33(2):130–138, 2009.

Acknowledgements

I would like to thank everyone who has supported me throughout my PhD. I would especially like to thank Dr. Roy Ruddle for his excellent supervision, and my colleagues in the Visualization and Virtual Reality Research Group for their comments, questions, and feedback on my research. Thanks to John Hodrien for sharing much useful technical knowledge during the development of my collaborative software, and to Chris Rooney for helping to ensure that the experiments ran smoothly.

Thanks also to Dr. Raymond Kwan and Dr. Vania Dimitrova for their help and guidance as postgraduate tutors.

This research was funded by a Doctoral Training Grant Studentship from the School of Computing.

Abstract

Collaborative virtual environments (CVEs) extend existing virtual environment (VE) technology to enable it to run over a network (e.g. the Internet), and introduce mechanisms that allow multiple people to co-exist, be aware of each other's presence (e.g. through avatars) and communicate. CVEs are useful for when teams of people want to collaborate when they are geographically separated, e.g. in games [14], social communication [65], visualisation [120], computational steering [17], or alternatively people might be spatially collocated in the real world but wish to work together in a VE, e.g. military training [99].

The dream is for interaction in CVEs to be more effective than interaction in the real world. The increase in globalisation and geographically distributed personnel who need to collaborate, act as a driving force for the development of effective collaborative technologies, which would allow businesses to save time and money, help distributed communities stay in touch, and reduce the impact on the world's environment. The work presented in this thesis aims to make collaborative interaction in virtual environments more effective, more like that of face-to-face interaction, without unnecessarily restricting virtual collaboration to the naturalistic constraints of the 'real world' (cf. [79], [39]).

This thesis describes the implementation and evaluation of techniques to support synchronous and asynchronous collaborations in virtual environments. The techniques were evaluated in the context of an urban planning application, where proposed developments could be modelled in 3D and evaluated by members of the public (and potentially clients, architects) to decide if they support or object to the designs (e.g. [30]).

Synchronous collaborations were supported by a suite of techniques called Mobile Group Dynamics (MGDs), which were introduced and evaluated in two stages (Chapters 4 and 5). First, a novel 'group graph' metaphor was used to explicitly show the groups that people had formed themselves into (and help people locate the whereabouts of their collaborators), and techniques were provided to help people move around together and communicate over extended distances. The techniques were evaluated by providing one batch of participants with MGDs and another with an interface based on conventional CVEs. Participants with MGDs spent nearly twice as much time in close proximity (within 10m of their nearest neighbour), communicated seven times more than participants with a conventional interface, and exhibited real-world patterns of behaviour such as staying together over an extended period of time and regrouping after periods of separation (Chapter 4).

Second, three additional techniques were introduced (teleporting, awareness and multiple views) which, when combined, produced a four times increase in the amount that participants communicated in the CVE and also significantly increased the extent to which participants communicated over extended distances in the CVE (Chapter 5).

Asynchronous working in CVEs was assisted using the metaphor of Virtual Time (VT), where the utterances of previous users were embedded in a CVE as conversation tags (Chapter 6). With VT, participants chose to listen to a quarter of the conversations of their predecessors while performing the task. The embedded conversations led to a reduction in the rate at which participants travelled around, but an increase in the live communication that took place. Taken together, the studies have implications for CVE designers, because they provide quantitative and qualitative data on how group dynamics functioned in a CVE, and how synchronous and asynchronous groupwork was improved by using MGDs and VT techniques. In addition, the rich complexity of possible functionality for VT highlights a number of possibilities for future research.

Contents

List of Figures

List of abbreviations

API	Application programming interface
AR	Augmented reality
CAVE	Cave automatic virtual environment
CSCW	Computer supported cooperative work
CVE	Collaborative virtual environment
DIS	Distributed interactive simulation
HMD	Head-mounted display
HUD	Head-up display
IDE	Integrated development environment
IM	Instant messaging
IP	Internet protocol
LAN	Local area network
MGDs	Mobile group dynamics
MMORPG	Massively multiplayer online role playing game
NLP	Natural language processing
PDU	Protocol data unit
TCP	Transmission control protocol
UDP	User datagram protocol
VE	Virtual environment
VoIP	Voice over Internet protocol
VR	Virtual reality
VT	Virtual time
WYSIWIS	What you see is what I see

Chapter 1

Introduction

Virtual environments (VEs) are three dimensional electronic worlds that contain information (e.g. 3D design models) that a user can interact with (e.g. visualise, navigate, modify). VE technologies have many application areas, e.g. engineering [57], medical visualisation [25], urban planning [30].

Collaborative virtual environments (CVEs) extend VE technology to enable it to run over a network (e.g. the Internet), and introduce mechanisms that allow multiple people to be co-present in the virtual environment. CVEs are useful for when teams of people want to collaborate. The people may be geographically separated, e.g. in games [14], social communication [65], visualisation [120], computational steering [17], or they might be spatially collocated in the real world but wish to work together in a VE, e.g. military training [99]. CVEs are introduced in more detail in Section 2.2.

The work presented in this thesis aims to improve collaborative interaction in virtual environments, and this research fits within a more general framework of collaborative interaction, introduced in the following section. In addition, examples are provided of the usage of technology to aid collaboration. Section 1.2 gives the problem statement, and Section 1.3 outlines the hypotheses for this thesis. Section 1.4 gives an overview of the thesis as a whole.

1.1 Collaborative interaction

Collaborative interaction in general may be classified in terms of time (synchronous vs. asynchronous) and space (collocated vs. remote) [56]. Table 1.1 shows examples of collaborative interaction classified in these terms. For example, face-to-face 'real life' collaboration takes place at the same time and in the same place (synchronous and collocated). Technology (phones, video conferencing, computer supported cooperative work tools, collaborative virtual environments) widens the possibilities to include remote collaboration.

Asynchronous collaboration can occur in the same place (e.g. communicating by leaving post-it notes in a shared office), or in a different place (e.g. using email, voicemail, or a wiki).

Space \ Time	**Synchronous**	**Asynchronous**
Collocated	Face-to-face; Computer aided design; Virtual prototyping	Leaving notes/messages in a shared office: post-it notes, whiteboard; People passing on messages on behalf of others
Remote	Telephone; Video conferencing; Chat/social systems (e.g. Second Life); Online games (e.g. World of Warcraft); Collaborative visualization; **Mobile group dynamics (Chapter 4);** **Teleporting, awareness & multiple views (Chapter 5)**	Email; Voicemail; Newsgroups; Wiki; **Virtual time (Chapter 6)**

Table 1.1: Collaborative interactions classified in terms of time and space [56]

People have many modes of communication available to them in synchronous, collocated situations. Face-to-face communication includes eye-contact, body language, verbal communication, and multiple channels of communication (talking to one's neighbour vs. addressing the group as a whole). Further, communication can be supported by providing people with tools such as whiteboards, flipcharts, or digital technology (e.g. projectors, interactive whiteboards). On the other hand, providing team members with their own space reduces the chance of interruptions and distractions, and increases privacy. Lai et al. highlighted research that showed benefits at both ends of this scale: 'radical collocation' where team members were in the same room without cubicles (known as a 'warroom'), [114], vs. private workspaces [28], cited in [62].

Can technology provide an idealised scenario, where team members can benefit from the privacy of their own workspace, while technology provides awareness of their fellow team members' activities, and makes them available for communication? Can we provide an easy way to bring people together (metaphorically speaking) for effective, synchronous collaboration, and then allow people to switch to their own space, leave the group, move away, at the push of a button? This would be particularly useful in cases where team members are distributed, and meeting up face-to-face would be time consuming.

An example scenario where technology is being used to support collaboration in this way (i.e. collaboration between team members who are spatially separated) is in software development. Technology is needed to help team members share knowledge, but in addition it is important to build up social aspects of collaboration (the team members need to have cohesive relationships) [59], and to have awareness of other members' activities [114]. Technology platforms such as Jazz (from IBM Rational) [54], and Visual Studio Team System (Microsoft) [72] aim to provide these benefits.

Cheng and coworkers engineered Jazz, for which they envisioned a growing trend of geographically distributed software development teams [23]. The Jazz project integrated communication and collaboration technologies into the 'Eclipse' integrated development environment (IDE) [32]. The project combined synchronous and asynchronous collaboration technologies, e.g. instant messaging (IM) and email communications. It provided awareness of other team members' activities by allowing users to see photos of the people who were online and coding, and provided functionality for status messages—team members could provide a one or two sentence description of what they were working on, similar to status messages used in IM, and on social networking sites such as Facebook [34].

The Jazz project and similar collaborative tools are from the field of computer supported cooperative work (CSCW). Collaborative virtual environment (CVE) technologies are a subset of CSCW tools, and are introduced in Section 2.2.

An extension to Jazz that used CVE technology was called Bluegrass [85]. In Bluegrass, teams were allocated with a plot of land in a 3D environment. Team location, work items, and meeting places were represented visually by objects in the environment, such as trees, gazebos and patios. The system was linked to IBM's social networking system, known as Beehive, and provided information in the environment from users' profiles. Does the social aspect of collaborative virtual environments increase team cohesion, which in theory would enhance productivity [58]? The evaluation of collaborative work can help answer this question, and Section 2.5 discusses this in more detail.

Table 1.1 categorises the research in later chapters. The work is placed in the remote category (users are spatially separated). However, there is a difference between CVEs and conventional CSCW tools: the former allow people to be virtually collocated (they provide a 'place' for users to meet up and interact [47]) when they are physically remote.

The research from Chapter 4 describes a suite of techniques called Mobile Group Dynamics (MGDs), which helps people work together as they travel around large-scale virtual environments. Chapter 5 improves these techniques by adding teleporting, awareness of who is within hearing range, and multiple views of the environment. Finally, Chapter 6 describes implementation and evaluation of a paradigm for asynchronous collaborations in virtual environments: a system called Virtual Time (VT).

The following sections introduce the problem, hypothesis, and provide a brief overview of the thesis as a whole.

1.2 Problem statement

It is well known that collaborative interaction in virtual environments is non-trivial, e.g. due to problems with perspective [51], usability difficulties [7], and a lack of awareness of the activities of others [75]. Section 2.3.2 discusses these problems in more detail, and provides more references to related work.

The work in this thesis aims to improve collaborative interaction in virtual environments for synchronous and asynchronous remote collaboration (Table 1.1). Synchronous work is facilitated by a suite of techniques called Mobile Group Dynamics (MGDs). MGDs support 'group dynamics' (the processes by which people form themselves into groups and operate), as people travel around (i.e. are mobile) and work together in a large-scale space. Asynchronous work in virtual environments is assisted using the metaphor of Virtual Time (VT), where participants have access to conversations and activities of those who were in the environment before them.

The following chapter introduces CVEs, details their technology and applications, and gives a literature review on group dynamics. Following this a scenario is given for the evaluation of the MGD and VT techniques. Chapter 3 details the technical implementation of the CVE used for evaluation, and the experiments themselves and the individual hypotheses are given in the respective chapters. MGDs are introduced in the thesis in two stages: Chapters 4 and 5, and VT is detailed in Chapter 6.

1.3 Hypothesis

The overriding hypothesis of this research was that MGDs and VT would improve synchronous and asynchronous interaction in CVEs, respectively. This hypothesis is broken down to show the predicted outputs of the research (below). The metrics in this research are the communication between participants, the spatial positioning of participants within the virtual environment, the usage of MGDs and the usage of VT. More detail of the latter two metrics is provided in Chapters 4, 5 and 6.

An improvement of synchronous teamwork taking place in the CVE was expected to be identified by two major outputs from the evaluations of MGDs. The first output was predicted to be a greater amount of conversation about the task. More communication in general can be attributed to lack of sensory information, e.g. making the implicit explicit [51]. However, in the research presented in this thesis participants were provided with greater sensory information (information provided by the MGDs techniques) and the communication was coded to identify task related utterances for analysis. The second output was predicted to be closer spatial proximity. Participants working primarily as individuals were expected to navigate separately and times spent collocated would be coincidental. In contrast, participants working together as a team were expected to exhibit closer spatial proximity as they engaged in collaborative navigation and shared perspectives to share ideas.

The outputs of the usage for MGDs and VT related more specifically to their implementation. In summary, they were the usage of explicit groups (how they were formed, how many people were a part of a group, and the number of groups formed), and the usage of MGDs to collaboratively navigate and collocate within the environment. These outputs are discussed in context in the corresponding chapters (Chapters 4 and 5).

The hypothesis for the second stage of MGDs (Chapter 5) specifically builds on the results of the first: the awareness and multiple views functionality were predicted to increase the distance over which participants communicated, and more details of this are given in the related chapter.

The hypothesis for the VT study was difficult to generate, since there are few observations of asynchronous teamwork in virtual environments. The initial rationale was that conversation tags would allow participants to benefit from the work of previous users, and so the metrics were the usage of the conversation tags, the performance on the task and the distance travelled. These are outlined in more detail in Chapter 6.

1.4 Overview

This chapter has introduced collaborative interaction in general, and provided some examples of computer supported cooperative work (Section 1.1). A taxonomy classified various methods of collaboration, and showed how the work in this thesis fits within that framework (Table 1.1).

Chapter 2 introduces collaborative virtual environments in more detail. This includes a description of the many ways these can be implemented (technology, software and hardware, network architecture), and the application areas are provided with reference to related work. A literature review follows on group dynamics, (the forming of groups and performing of activity), and a background is given on the new concept of virtual time (VT: a paradigm for asynchronous interaction in CVEs). The background is given on CVE evaluations, and the scenario used for experimentation in later chapters is provided to set the scene.

Chapter 3 details the technical implementation of the CVE software used for the evaluations in subsequent chapters. Novel techniques to support synchronous interaction are evaluated in Chapter 4 (MGDs), and two main areas for improvement are identified in Chapter 5. New functionality (teleporting, awareness and multiple views) is evaluated to address these issues. Chapter 6 explains the concept of virtual time (VT), and discusses its implementation and evaluation. Finally, overall conclusions to the research are drawn in Chapter 7, and possibilities are outlined for future work.

Chapter 2

Background

2.1 Introduction

This thesis describes techniques to support synchronous and asynchronous teamwork in virtual environments (VEs). The work fits within a more general framework of collaborative interaction, communication and working, outlined in Section 1.1. The following sections outline collaborative virtual environments (CVEs), the technologies (hardware and software) and techniques used to generate and interact with CVEs, the applications, and ethical issues. Following this, a background is provided for group dynamics, which takes Tuckman's model of group processes (Section 2.3) and uses it as a framework for a discussion of related work in supporting group dynamics in CVEs, and the different possibilities for group interaction. This is followed by an introduction to asynchronous collaboration, and methods for evaluating groupwork. Finally, the scenario used for the evaluations in later chapters is introduced.

2.2 Collaborative virtual environments

Collaborative virtual environments (CVEs) are three dimensional electronic worlds that combine shared information (e.g. 3D design models) with mechanisms that allow multiple people to co-exist, be aware of each other's presence (e.g. through avatars) and communicate. CVEs allow collaboration to take place within the context of a large-scale 'space'. Section 2.2.4 details the applications of CVEs, and Section 2.2.6 defines large-scale space.

CVEs can use standard desktop monitors with ordinary peripherals for interaction (mouse, keyboard), or alternatively they can use sophisticated immersive virtual environment technology. The experimental research reported in this thesis used desktop systems, although the principles could be applied to immersive technology in future.

More detailed descriptions of VE technologies can be found in [40] (including a chronology) and [16] (3D interaction), and an overview of the field as a whole can be found in [108].

2.2.1 Interface devices

CVEs can use ordinary peripherals for interaction such as a mouse and keyboard, or a gaming control pad. A typical gaming control pad (e.g. PlayStation 3, Xbox 360) will have two analogue sticks, a digital direction pad and numerous input buttons.

A glove interface wraps around your hand to allow interaction. One of the earliest examples of a glove interface is from Zimmerman et al. in 1987, [127]. Their gloves tracked position and orientation of the hand, and the flexing of the fingers. In addition, tactile feedback was provided when the user's virtual hand overlapped a virtual object. Not long after, Fakespace developed pinch gloves, which could detect combinations of the wearer's fingers and thumbs pressed together [35].

Haptic interaction devices use the sense of touch. This could be achieved using tactile feedback (see above), an exoskeleton allowing measurement and restricting of hand/arm movement (e.g. [42] [60]), or a robotic arm controlling force feedback (e.g. the Phantom from SensAble Technologies [98]). A brief history of haptic technology can be found in [111].

Motion tracking can be performed using a variety of techniques, including magnetic tracking, inertial tracking, and optical tracking [16]. Examples of devices used for motion tracking are Ascension's Flock of Birds magnetic tracking system [5], the Precision Position Tracker from WorldViz [119], Vicon's optical tracking [117], the Moven inertial tracking suits from Xsens [121], and the Nintendo Wii Remote designed for the games industry [76]. There are also devices to track eye movement, such as the EyeLink from SR Research [107].

A touchscreen display can allow for interaction using hands or a stylus, e.g. a Tablet PC, the multitouch 'city wall' project [80], which is useful for naturalistic manipulation of objects.

2.2.2 Interaction techniques

Interaction techniques in general are designed and evaluated with respect to a particular subset of interaction requirements (e.g. selection, positioning), or for application specific tasks [16].

For example, selection tasks can be achieved using ray casting techniques, where the position and orientation of a user's hand is tracked, and a ray is drawn originating from the hand and extending along the direction it is pointing. Objects that intersect with the ray are selected. Positioning can be done using a virtual hand, where the users hand is tracked and rendered into the environment, and moving objects is done in a naturalistic fashion, or alternatively using a function of the coordinates of the real hand (e.g. positioning the virtual hand twice as far away from the body as it is in reality to allow the selection of distant objects).

A detailed account of interaction techniques for a variety of VE platforms and technologies can be found in [16]. The research in this thesis focused on interaction using desktop CVEs.

2.2.3 Software and network architecture

An interactive VE application receives input (e.g. from devices outlined above) and generates output to the graphics hardware for visualisation, using techniques from the field of computer graphics (e.g. texture mapping, illumination models). An overview of the field of computer graphics can be found in [37], [48], and the latter demonstrates how to produce graphics using OpenGL, an industry standard application programming interface (API) specification.

A CVE system extends VE technology (graphics, input devices) to enable interaction across a network. Notable CVE implementations include SIMNET [99], NPSNET [67], MASSIVE [43] and DIVE [20].

This section focuses on design decisions common to all collaborative (networked, multiuser, shared) virtual environments, specifically the network architecture and the methods used for maintaining a shared state (cf. [99, ch. 4 and 5]).

The network architecture can be peer-to-peer or client-server based. SIMNET was one of the first collaborative virtual environment systems, and it used a peer-to-peer architecture [99]. Each active object in the environment (e.g. vehicle, destructible scenery) was controlled by a different host machine, and each host transmitted the state of its object using Ethernet multicasting (which reduced network traffic compared to point-to-point communication). No single host maintained the overall 'master' state of the environment.

This architecture has the advantage of being fault-tolerant (host-failure does not bring the whole system down), but it creates difficulties when objects interact. For example, one host may perceive that a collision between two objects has taken place, and another may not. A more complex scenario would be when two hosts detect slightly different versions of the same collision event, where each version would lead to a different outcome. For example, consider two vehicles colliding and a physics engine that generates realistic responses. If two hosts detected the collision occurring with slightly different parameters (coordinates, orientation, velocity) then the vehicles would end up in different positions, thus affecting their subsequent interaction with other objects (e.g. other cars on the road). This is a major problem for the system designer.

One solution would be to change the network architecture. A client-server model creates a hierarchy, where the server maintains the shared state of the environment. In an ideal lag-free world each host could transmit its input to the server (or primary host), which would change the state of the environment accordingly and respond with the new state. This would avoid any conflict between hosts. Object interaction can be dealt with using a locking mechanism, so that only one host can update an object at a time. However, in reality this is not very scalable, since it places greater load on the network (all updates and object locks need to be transmitted) and the network lag delays feedback for the user.

Designing the architecture to maintain consistency and allow dynamic interaction is a balancing act known as the 'consistency-throughput trade-off' [99, p. 102]. On the one hand we have a system that maintains absolute consistency, achieved by transmitting all events to the server and waiting for a response that confirms the new state of the environment. This requires methods to reduce the network load, to maintain a fast response rate and make the system scalable. DIVE deals with this by sub-dividing the environment into smaller sections which are only communicated to a small number of subscribers [41]. The network load is further reduced by using multicast.

On the other extreme is a system with a high frequency of state changes and a fast response to user input. Such a system cannot wait for each state change from the server, and typically not all changes are transmitted. Each host attempts to predict the current state of the environment based on infrequent updates from the server. For example, predictions of object locations can be based on their initial direction and velocity (a procedure known as 'dead-reckoning' [99]). This type of system can provide direct feedback to the user (when they press a button on a movement control device, they move straight away without waiting for a response from the server). Local copies of objects can continue their movement/behaviour during the time gap between state updates. However, such a system must be able to deal with each client having a different local representation of the environment. The longer the time gap between state updates from the server, the more the clients' representations will differ.

Many CVE systems use a mid-way approach, allowing each client to have their own local copy of the environment (faster response to user input) and having frequent state updates from the server (better consistency). In order to do this, extra methods have been developed to help reduce the network load. For example, MASSIVE [43] uses a spatial model so that each object has an aura (how visible they are to others) and a nimbus (their area of interest). A source object only transmits to a destination object when its aura intersects with the destination's nimbus. MASSIVE-3 [46] deals with inconsistency between hosts by using sequence numbers for events, including separate counters for non-important events that do not require guaranteed transmission (e.g. position updates). The transmission of events also includes a list of sequence numbers that must be processed before the new event can be carried out (otherwise it might not make any logical sense in the out-of-date environment).

There is a special case where allowing each host to maintain a local representation of the environment can cause further problems. This is when users perform shared object manipulation, and they manipulate the same attribute of the same object. Concurrent object manipulation is beyond the scope of this research, but methods to deal with this can be found in [82] (academic CVEs) and [31] (using techniques from the games industry, from libraries that have recently been released as open source).

2.2.4 Applications

CVEs have a wide range of applications, including social communication, entertainment, visualisation, military training, and product design.

CVEs are used for social communication in systems such as Second Life [65] and There [70]. Second Life is an environment that uses standard desktop technologies connected via the Internet. It allows user-generated content, such as the designing and scripting of objects (e.g. clothes, vehicles). Participants can buy and sell items, including land on which they can build. There are no specific objectives in Second Life, but one has the freedom to explore, meet people and even set up a business earning virtual money that has real world value.

Online games use similar CVE technologies to Second Life, but the content would usually be created by full-time system developers (as opposed to the participants themselves), and they have a measure of progress, e.g. the level system in World of Warcraft [14]. World of Warcraft is a massively multiplayer online role playing game (MMORPG).

Participants pay a subscription fee to use the game. They can form teams (guilds) and work together for fun or to help progress in the game.

Virtual environment technologies are used in a subset of visualisation work. For example, medical data acquired from MRI or CT scans can be visualised in a 3D environment. A series of images are generated from the scans, and these are grouped together to make a three-dimensional ‘volume’ data set. Each discreet point in this data set is called a volume element, or voxel. These voxels are rendered onto a 2D display using computer graphics techniques such as volume rendering (each voxel is assigned a colour and alpha value and rendered to the display), or an ‘isosurface’ (data points of similar properties are joined up to create a 3D contouring surface). For example, Cohen and colleagues used volume rendering to visualise medical data, and provided neurosurgeons with a tool to help them understand the nature of brain aneurysms [25].

Some visualisation applications benefit from collaborative and distributed interaction (i.e. interaction across a computer network). Consider a scenario where a patient needs urgent diagnosis, but the limited number of qualified personnel (or simply the geography of the country) means that waiting for a medical consultant to arrive in person takes too long. Medical decisions could be made using collaborative systems (e.g. a shared volume rendering). An example of a collaborative visualisation system is [120].

Another example of collaborative visualisation is the pollution demonstrator (proof of concept) from Brodlie et al. [17]. This visualises pollution spreading across a map of the UK using a grid-enabled IRIS Explorer (visualisation system) application. The concept is one of disaster control, where the path of a pollutant is predicted and decisions are made as to where evacuations need to take place. The simulation parameters can be modified based on data provided by a meteorologist, and this is done by a scientist while the simulation is in progress (a process known as ‘computational steering’). The steering and visualisation operations are done collaboratively, so the scientists, meteorologists and decision-making officials involved do not need to be collocated (distributed collaboration saves valuable time).

CVEs have been used for many years for military training [99]. A real or hypothetical environment is mapped out in 3D, and military personnel are given a variety of simulated operations to perform. The collaborative nature allows multiple people to learn to work together as a team in a safe environment, before being deployed in the real world.

In the field of design, VEs allow the ‘virtual prototyping’ of products before they are physically built in the real world. This helps with problems such as manufacturing variation, where material properties and manufacturing process mean that measurements on the final product do not always conform to exact specifications. This application is tackled by Juster and coworkers in [57]. They begin with a tolerance level: the amount of acceptable manufacturing variation. Their software allows designers to visualise the worst-case, best-case and most likely outcomes of different tolerance levels in the design, and can then work out the trade-off between tolerance and cost (a low tolerance level is more likely to produce aesthetically pleasing results, but will increase the cost of manufacturing).

2.2.5 Ethical issues

Social systems and games are used by large numbers of participants who spend a substantial amount of time connected to the environments. Blizzard Entertainment released a press statement at the beginning of 2008 saying that it had reached ten million subscribers to World of Warcraft [13] (note that this does not represent the number of players that are simultaneously connected). Online surveys ran by Yee, [123], collected data on the usage of MMORPGs over a three year period, and the results showed an average usage of 22 hours per week among respondents.

These data highlight ethical concerns with the design of entertainment CVEs. For example, if large numbers of people are using these systems for long periods of time, it is important to consider and study the effects of these environments on real world behaviour. It is known that some psychological and behavioural changes induced in the virtual world have transferred to the real world in lab-based studies [125] (see Section 2.3.2.1). To what extent do social and gaming environments induce certain types of behaviour on participants? Perhaps a more common question would be 'do games cause violent behaviour in the real world?' However, progress on this is likely to be as slow as other areas (film, television), due to the number of parameters involved [95, pp. 710–711].

Environments like Second Life raise some social, ethical and legal issues, due to the virtual currency having real world value. Users can create objects that can be bought and sold. This leads to people working in the environment for real money with a grey area of legality, and throws up issues of ownership rights and the responsibility of the system developers to maintain suitable software security [68].

Ethical issues also occur in environments with a small user base. Milgram's controversial obedience experiment was replicated in a virtual environment, in which users had to give electric shocks to a virtual human. The results showed that users responded with distress even though they knew that the avatar receiving the electric shocks was not a real person [101]. The motivations for some experiments may be more positive, such as one designed to provide help with social phobia (e.g. [78]). However, with knowledge that the environment may affect people (even though they know it's not real), care must be taken to avoid unhelpful side-effects, e.g. could the exposure to the virtual environment have negative effects on the participant? Some participants felt so uncomfortable in the virtual Milgram experiment that they used their right to withdraw. If the 'real' Milgram experiment can not be carried out due to ethical concerns, then should we be allowing virtual Milgram experiments when the participants are reacting as if it is real?

The study of collaborative interaction presented in this thesis focused on groupwork in virtual environments. It used a custom environment developed by the author and participants gave their informed consent in writing prior to taking part. The environment was not designed to be uncomfortable to participants in any way, and the studies were approved by the Faculty of Engineering Ethics committee.

2.2.6 Large-scale space

Large-scale spaces are those in which 'Multiple vantage points must be occupied in order for the space to be visually apprehended in its entirety.' [118, p. 42]. An example of a large-scale space is a city or a building. In contrast, small-scale space can be apprehended by rotating the field of view from any vantage point, e.g. a room or a park.

Historically, the majority of CVE research used small-scale spaces (e.g. [9], [51], [102], [66]). An example of early work designed to accommodate large-scale interaction is [44], which worked to reduce network bandwidth by changing the representation of groups of objects (see section 2.3.1.4). More recently, online social environments and games such as Second Life and World of Warcraft (Section 2.2.4) provide large-scale interaction and researchers have used these environments to study behaviour (e.g. social norms in Second Life, [126], and avatar manipulation in World of Warcraft, [73]).

Large-scale space introduces extra challenges for groupwork, because not only do individuals get easily disoriented when they navigate a large-scale VE, it is also all too easy to lose track of the whereabouts of one's collaborators. Therefore, the related work in the following sections (the forming of groups and the performing of activity) is considered with respect to the size of the space and the number of collaborators involved.

2.3 Group dynamics

Collaboration has long been studied within a socially driven context in real life (group dynamics, [64]). The model of *forming*, *storming*, *norming* and *performing* has been constructed to describe group processes [116]. Storming and norming are the processes by which individuals' roles within a group become refined, whereas forming and performing govern the creation of groups and their ability to do work. It is these latter two processes that are most relevant for supporting groupwork using technology: how are groups formed and how do they perform activity?

2.3.1 Forming

Four key points about group formation need to be considered. These are the method of joining (implicit vs. explicit), how members are identified, the structure of the group (e.g. subgroups/hierarchy), and the way that the group is represented (e.g. aggregate views of the group as a whole).

2.3.1.1 Method of joining

When people meet and communicate informally in the real world they gather together into circles to hear each other. The groups are organised using spatial positioning so membership is implicit, and social etiquette applies when people join or leave. For example, new members may be invited to join by existing members' body language (e.g. stepping back to allow a newcomer into the circle), and when members leave the group they would often give an appropriate verbal indication or gesture (e.g. say or wave goodbye).

Active Worlds [2] is a chat-based CVE in which users form implicit groups. If users are too far apart, the chat text isn't displayed, so they are forced to gather together into rough circles to 'hear' each other. Groups can make themselves open to new members by gathering around the entrances to the worlds, or groups can govern themselves by agreeing a time and place to meet. The environments are large enough for this to provide privacy from users who were not invited because a group is unlikely to be found by accident. However, a disadvantage of this implicit approach is that the system maintains no record of the makeup of each group, so members may be unaware if they met by chance in another part of the CVE. Furthermore, forcing users to collocate into groups limits the scalability as large groups would clutter the visible space.

On the other hand, people can be part of explicit groups, for example a guest list for a wedding, a university society or sports club. Explicit groups maintain a formal record of their membership. In some cases membership is open (any student can join a society, they just need to sign up) but in others it is dictated by members who have special privileges (e.g., a couple deciding who they will invite to their wedding).

Social networking sites such as Facebook [34] use explicit groups. Membership can be decided by a group administrator (as in the guest list example) or it can be open to anyone (as in a society or club). Second Life [65] and There [70] are chat-based CVEs that implement explicit groups in a similar way to social networking sites. They use a menu based interface to allow the forming and joining of groups for people with a particular common interest.

2.3.1.2 How members are identified

'There' has no way of identifying who belongs to one of the groups by using the 3D environment (e.g. from the appearance of users' avatars). The only way to identify members is by consulting the group membership lists. Second Life provides limited help by writing group member 'titles' beneath the avatar names (the titles are not predefined, but could be something like Treasurer or Secretary), which is useful if you are aware of the titles within a particular group.

Group members can also be identified by spatial positioning or colour schemes. The former approach is used by a second type of group in There. A group is started after one user has chatted with another for a short period of time and the camera automatically switches to a special 'chat mode' that shows the users' two avatars aligned side-by-side, giving a better view of the group. New members can join by walking up to the group, clicking on an icon associated with it, and selecting the join option from a menu that appears. As more people join the avatars are arranged into a semi-circle, so each user can see all the group members in one view on screen (Figure 2.1). The disadvantages are that group members are immobilised for the sake of visual clarity, and only a small number can join.

A soccer team provides a good example of the real-world use of colour schemes. Membership is decided explicitly before the match starts, and it is communicated by the players wearing their team's colours. There is no question who is on which team, and it is straightforward to identify who is a member from a distance. A similar approach is used in entertainment CVEs such as Wolfenstein: Enemy Territory [106] where members of the two opposing armies can be identified from the uniform worn by users' avatars. Identification using colour schemes or uniforms is appropriate for scalability and means that people are free to move around the environment while both remaining a member and being identified as part of their group.

2.3.1.3 The structure of the group

Groups may change structure. For example, consider an office meeting. The people present may divide themselves into subgroups to carry out certain tasks, or someone may talk to the person next to them, using 'side-channels' of communication [12] rather than

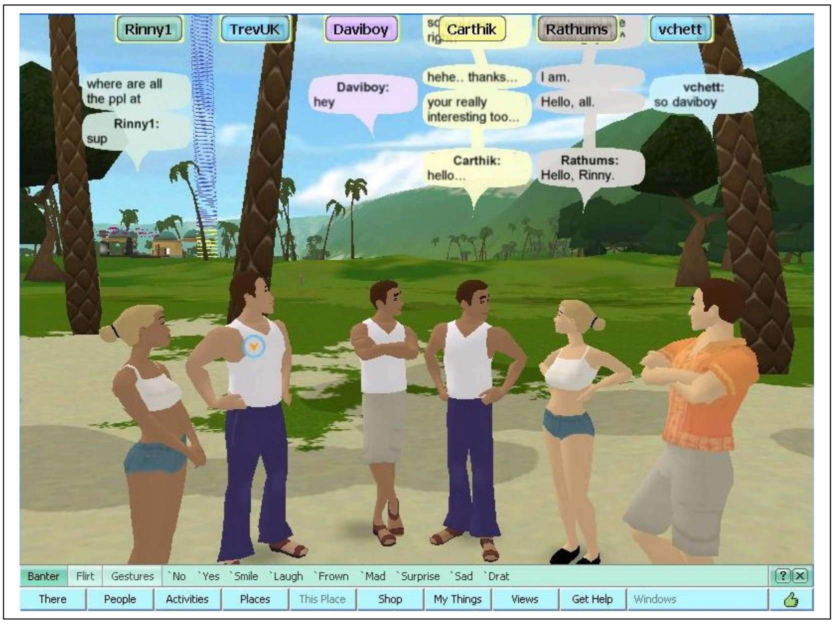


Figure 2.1: A group of users in There, a chat-based CVE. When users join together in a chat group, their avatars are positioned into a semi-circle, and the camera is automatically positioned to show all the avatars.

addressing the group as a whole.

There are three options for changing group structure. It can be (1) decided by the people involved (e.g. a product design team decide to divide up into subgroups and work on different parts of the product), (2) dictated by a leader (e.g. infantry divided up into ‘fireteams’ by their commanding officer), or (3) in the case of computer supported cooperative work it may be automatically carried out by the software application.

Investigations into the changing of group structure in CVEs are rare, but an exception is a study by Linebarger et al. that added functionality to a CVE to support explicit changes of group structure, and looked at the difference between manual and automatic sub-grouping [66] (see options 1 and 3 above). Subgroups could be formed using menu-based selection (manual), or by the system itself that formed subgroups when people were detected as working on different parts of the task (automatic). Automatic sub-grouping was seen to be disruptive (3), but manual sub-grouping was a success (1). The effects of option 2 in CVEs remains open to research, but if Linebarger’s conclusions are correct then it is the cognitive processes of participants breaking down the task and forming themselves into subgroups that are important, and these are not present in options 2 and 3, which each impose a divide from outside the group.

2.3.1.4 Representation

The way a group is represented can change. This is specific to CVEs. For example, MASSIVE-2 [44] implemented a concept of 'third party objects'—objects that affect the awareness between other objects (i.e. users). In their 'Arena' work, they used third party objects to hold crowds of users. Members of a crowd could see other individual members, but non-members saw an aggregate view instead (a large avatar). One application of this concept was to improve scalability (save on rendering and network bandwidth). However, groups were formed based on spatial positioning and so were limited to a small-scale space (they had to fall within the boundary of the third party object).

2.3.2 Performing

Performing is the stage in which a group carry out a task. The task can be performed by an individual, or a group of people collaborating together.

People carry out activity at an individual level by making use of the information available to them. An individual's awareness of the current state of the environment (known as their 'situation awareness') guides their decision making, which leads to the performance of actions, which in turn changes the state of the environment and the process repeats itself [33].

When people work together as a team, the objects in the environment and the team members themselves form a distributed cognitive process [52]. For example, in an airplane cockpit the pilot and co-pilot make use of the instruments available to interpret their situation. They need to communicate their comprehension of the information that surrounds them as they coordinate their activity. Section 2.3.2.1 gives details on communication processes.

In other scenarios, team members need to travel around the environment to perform a task, such as in military training. They still need to communicate in some way (e.g. radio)—communication is relevant to all collaborative activity—but in a large-scale space they need to move around the environment too. Section 2.3.2.2 discusses the issues involved with movement in the real and virtual world.

When a team has formed and begins to carry out activity, we want them to work together efficiently and effectively. Section 2.5 considers how group performance can be measured and what methods might be used to increase productivity.

2.3.2.1 Communication

When a group of people work together as a team, information is taken from the environment (e.g. objects and context, such as the position of the ball in a soccer game) and then communicated between the group members to form a behavioural response (e.g. passing the ball and scoring a goal). Communication can be verbal or non-verbal (different modes), and is influenced by appearance (of self and others) and perspective (one's own interpretation of the information available).

Mode When we communicate face-to-face in the real world, different sensory modalities work in parallel and using more than one makes interaction easier. Simple things like mimicking the body language of another participant increases social rapport in real and virtual systems [6]. Study of Second Life has found that real world social norms such as those found with eye gaze and interpersonal distance does transfer to CVEs [126].

The control interface for gestures and facial expressions has long been a problem in virtual environments [9]. Moore et al. describe the problem with respect to MMORPGs: 'the industry and the player base is already realising that most current massively multiplayer worlds place too much burden on the players' hands as they must be used to "talk," walk, gesture, interact with objects and access menus.' [73, p. 301]. The 'burden' on the users' hands can be reduced by using voice communication as opposed to text, and the vision is that one day we will have affordable real time motion capture for gestures (this is currently being researched and promoted in the games industry using computer vision techniques, e.g. PlayStation Eye, Nintendo Wii Remote).

Technology can overcome the 'natural' limits of real world verbal communication. For example, distance attenuation of verbal communication is removed when using a mobile phone. The Robust Audio Tool (RAT) used in the COVEN project [109] is the equivalent of a mobile phone for virtual environments, or a telephone conference when there are multiple users. This has limited scalability because the sound is 'broadcast' to everyone in the environment, and it does not allow for 'side-channels' of communication (i.e. private communication within subgroups, see Section 2.3.1.3 [12]).

A CVE can provide a real-world 3D distance model for audio, such as the Binaural sound model used by Tsingos et al. in [115]. The audio communication is mapped to the position of the speaker's avatar, attenuated by distance and given a calculated phase difference for each stereo channel, which is approximate to naturalistic verbal communication. The OpenAL API provides a simplified model of naturalistic communication, with distance attenuation and stereo sound [50]. The advantages of an environment using 3D audio include helping the user comprehend who is talking (because one can mentally map the source of the sound to the visual avatar), and reducing noise from multiple users by culling the distant sound sources, so that listeners only hear their neighbours.

Appearance There are two aspects important to appearance: how people look (clothes, embodiment) and how people behave (body language, eye contact).

In the real world people change their appearance (their clothes, hairstyle) to communicate, celebrate, and control how they are perceived by others. In the virtual world people want to control the design of their avatar [94], and this can be done in many existing environments (e.g. Second Life, There).

Avatar appearance has important implications for the user themselves. Yee and Bailenson constructed a virtual mirror, in which a user (who was in first-person perspective) could see their own avatar. It was found that changing this avatar changed the user's self-perception, which affected their behaviour in the environment [124] and also in the real world after the experiment [125]. For example, participants given taller avatars performed better in a negotiation task than participants with shorter avatars, and this transferred to subsequent face-to-face interaction. The research was extended to online games, where it was found that 'tall attractive avatars are most likely to be the highest level', [125, p. 25], where 'level' is the measure of progress in the game.

Avatars can be animated to create behavioural realism using body language and gestures, and this has been shown to have an effect on other participants in the environment. For example, Slater and Steed's virtual audiences reacted in positive or negative ways, which affected the participant giving a presentation [103]. Blascovich showed how an avatar's behavioural realism changed how close participants walked to the avatar. They didn't walk as close if it was behaving in a realistic manner, in a similar way to how we respect personal space in face-to-face interaction [11].

Perspective Participants in desktop VEs 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 [51]. (2) What you see is *not* what I see, which makes it difficult to understand another's perspective.

Problem 1 is likely to happen because of a narrow field of view. This is inherent in desktop VEs wanting to minimise distortion to keep realism [39]. Problem 2 is due to a removal of real world sensory data, such as eye movements and depth perception. This problem tends to be compensated for by a large increase in the amount of verbal communication that takes place [90].

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 [122, 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).

Hindmarsh and colleagues experimented with extending the field of view to help with problem 1 [51]. They used 'lenses' either side of the viewport that rendered a distorted view of the peripheral environment. They made the view frustum of participants visible for everyone to see to help with problem 2.

Murray and Roberts made an avatar's eye movements match those of the participant by using an eye tracker, so that others could determine their gaze direction [74]. This is a useful solution for problem 2 when using large displays. The eye movements of participants would lose their meaning in all but a small range of values with small displays, where large values would imply they are looking outside the display at something in the physical world.

Wössner et al. took a different approach in [120], and provided a What You See Is What I See (WYSIWIS) view in their CVE, which would eradicate problem 2 entirely. 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 didn't interfere with the other participant. Sonnenwald et al. found that users saw a benefit in both independent views and shared perspectives [105]. Users liked to be able to figure things out on their own and then discuss them collaboratively.

A WYSIWIS view shows the environment from another participant's perspective. However, even if two participants were stood at exactly the same point in the environment, there is no reason it should look the same. Technically speaking, the environment could be rendered differently to each participant, providing a subjective representation. Jää-Aro and Snowdon, in [55], argue that CVEs should make use of this. They suggest changing information for each participant (e.g. translating text into the participant's own language) and filtering content (e.g. reducing clutter in a CAD model by removing some information that the user does not require). Additional information can be provided for expert users. For example, an urban planning application could provide all participants with 3D representations of proposed developments (so members of the public could view them), and provide additional information to planning officials, such as 2D diagrams showing the layout of the proposed buildings. These diagrams could be integrated in the 3D environment itself, overlaid on top of the viewport, or positioned next to the viewport to augment the original display.

The key requirement for solving problem 2 is being able to easily interpret what others are seeing, and Smith et al. explain this as the 'What You See Is What I *Think You* See' principle, WYSIWITYS, [104]. This implies that participants can have their own independent view and auxiliary information that is unique to them, provided that others can come to a logical conclusion as to what they are seeing.

2.3.2.2 Movement

Movement around an environment in the real world can be individual (people split up and divide the task between them), as a group (to get a shared understanding), or require meeting at a point of interest [122]. Moving as individuals or as a group both have their advantages. Dividing the environment up between group members is a quicker way of covering the space, but navigating together allows the sharing of ideas—'two heads are better than one'—and mistakes in the task are less likely to go unnoticed. The group can take a hybrid approach and divide into subgroups, increasing speed of task performance and still benefiting from a small amount of groupwork.

Textures and landmarks can help individual navigation in the virtual world [63]. In addition, navigational aids help people know where they are in a large-scale space. Most used is simply a map, using visual momentum to indicate one's momentary position and orientation [4] [89], but bird's-eye views are also effective for seeing beyond one's immediate surroundings [27].

Moving as a group in a virtual world is a non-trivial task, due to the small field of view in desktop environments (it is easy to lose track of where other users are). An over-the-shoulder perspective helps when compared to a first-person perspective: users can see others relative to their avatar [22]. However, moving the camera behind the avatar just provides a bit more context, not a larger field of view, and difficulties still occur [51]. One solution is to use an abstract device to provide an indication of where others are (e.g. a radar, or 3D arrows pointing to targets [24]).

Moving to a point of interest in a virtual world presents communication and navigation problems. First, there is a combination of problems 1 and 2 from Section 2.3.2.1—when someone says 'come here and look at this', you need to know where they are, and what they are looking at. Second, you need to navigate to the point they are looking at, which will be helped by navigational aids outlined above.

2.4 Asynchronous collaboration

People benefit from working together *asynchronously* in real life. For example, someone who is out of their office can be sent a message by leaving a note on their desk, or a letter in their pigeon hole. Some types of asynchronous working are supported with technology, using tools such as email, wiki, voicemail. In any type of asynchronous system, participants don't need to be working at the same time, they have time to think about their decisions before responding, and information can remain for future reference.

Traditional CVEs bring together people for synchronous communication (they are logged onto the system at the same time), but by recording users' movements and conversations and providing access to this for future participants, CVEs can be extended to provide asynchronous, remote collaborations (Table 1.1).

There are few examples of asynchronous collaborations 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 [10], 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 [45].

Chapter 6 describes a general framework for asynchronous collaborations in VEs. A new concept is introduced called virtual time (VT), which is designed for asynchronous working, and a description of how VT fits within the general framework is given. (The framework also highlights other possibilities for the design of VT.) Finally, an empirical evaluation of VT is carried out and the results are analysed.

2.5 Evaluating groupwork

The analysis of groupwork can be divided into two categories: taskwork and teamwork [7]. Taskwork measures the productivity of the group with regard to the task they were given, and teamwork describes the group processes involved in achieving the productivity. For example, a group's high performance may be due to a minority of individuals (as opposed to the result of them all working together), or on the other hand a group's low performance may be the result of poor team cohesion (as opposed to substandard team members). The following sections give examples of the two types of work, and how they can be evaluated.

2.5.1 Taskwork

Most tasks have some measure of performance. For example, a football game (goals scored, position in the league), an exam (marks awarded), or a dance contest (judges award points). Some tasks *require* teamwork (football), other tasks don't, such as a business project that could be collaborative or alternatively could be carried out by people working individually. With the latter type of task, one would expect a benefit of people working together ('two heads are better than one').

However, as Kerr and Tindale discuss in [58], groups usually fall short of their potential in terms of performance. They note that group performance loss is due to problems such as poor resource management (e.g. not making use of the members with the most expertise), and social loafing, where individual group members do not pull their weight in the group. According to their review, efforts have been made to tackle poor resource management by helping people identify the group's most expert members (e.g. by providing performance feedback, or training as a group), and these ideas could be implemented in CVEs.

Social loafing is a well-known problem in organisational psychology, [58], and has been found to occur in technology supported teams, e.g. [113]. If individual effort is not recognised, then individuals can get away with less effort and have less incentive to increase their effort. One solution, therefore, is to recognise individual contributions, but that is not always easy. For example, it is hard to work out which football players contributed the most to a team's success (it is not necessarily the ones that scored).

Previous work in CVEs has measured performance using time taken to finish the task, [97] (solving a puzzle), [83] (building a gazebo). Another performance measure has been the quality of the final product in design or repair tasks. The way of determining the level of quality depends on the task. For example, Linebarger et al. told participants they were measuring the quality of their roller coaster designs by the number of loops and hills which matched the specification provided [66]. In another example, this time from the field of CSCW, Kraut et al. measured the quality of a physical bicycle repair job using a checklist of requirements [61]. Measuring performance is straightforward in problem solving or quiz type tasks, where the performance can be scored using the number of correct answers (e.g. a comparison between chat room collaboration and face-to-face is made using a problem solving task in [92]).

An analysis of individual performance may help us make predictions of team potential, e.g. [49] (comparing individuals and pairs of users), but the complexities of teamwork make predictions difficult (teammates could potentially be a hindrance rather than a help). Further analysis is required to give us insight into the teamwork taking place, to determine ways to make teams more efficient and effective when working in CVEs.

2.5.2 Teamwork

Schroeder and colleagues suggest quantitative and qualitative methods for the analysis of teamwork [96]. For example, Section 2.3.2 discussed how communication and movement are the building blocks for performing activity in a large-scale space. A qualitative analysis of communication would be done by observing patterns and trends in the choice of words and the use of language, cf. [91]. An analysis of the movement behaviours of participants in a CVE would also be qualitative, e.g. looking for typical patterns of movement, such as periods of time where people navigate together versus times when they split up and each explore separate parts of the environment. An example of a quantitative method is an analysis of the amount of communication, i.e. the number of utterances that took place. The utterances could be coded based on the topic of conversation (e.g. greetings, task-related or idle chat), and the number of each topic could be used in a quantitative analysis to get an understanding of what people were talking about.

The combination of qualitative and quantitative methods takes the raw data and draws up a picture of the teamwork that is taking place. Within this there are two levels of information that is useful to us: (1) what people were doing, and (2) why they behaved in this way (taken from levels 2 and 3 in [88]). The first level is an explanation of how people used the system, and how they went about performing the task. The second level gives a more in depth analysis into the reasons behind the participants' actions, and looks at the meaning of the participants' behaviour. For example, participants in a CVE might spread out and navigate the environment separately. This could be seen in a qualitative analysis (observation), and then quantified by computing the distance from each participant to their nearest neighbour within the environment (this could be calculated programmatically for the full duration of the study). This quantitative analysis would provide information about what people were doing in the environment (level 1)—it would report that they were spread out and the quantification would allow a comparison to other groups of participants. However, this does not become useful until the question of why they were behaving this way is considered (level 2). It might be discovered from a conversation analysis that they agreed to split up and divide the task due to a strict time constraint. Or, there might have been very limited communication between the participants, which would indicate that they were spread out because they weren't working together—they were actually performing the task as individuals and very little teamwork was taking place.

Evaluations of teamwork in related studies apply similar principles. For example, Slater et al. used questionnaires to evaluate the effects of asymmetric technology in CVEs (desktop versus HMD). They found that the participant using the HMD became the group leader [102] (a level 1 analysis). A more in depth analysis is required to determine why this happened despite the desktop participants not knowing that someone else was using immersive technology. Hindmarsh et al. carried out a quantitative analysis on the communication in their CVE, and the amount of conversation was relatively high (level 1) [51]. A further qualitative analysis looked at what the participants were saying, and determined that the reason for the large quantity of communication was due to lack of sensory information in the CVE when compared to real life (level 2)—participants were compensating for problems like narrow field of view by explicitly stating things that would be obvious in the real world (‘making the implicit explicit’).

2.6 Scenario

The research in this thesis developed techniques to facilitate teamwork in CVEs. An urban planning scenario was created for the techniques to be evaluated. There were other possibilities for the scenario that could have been used, e.g. virtual tourism, emergency evacuation/response (e.g. fire safety training), or a gaming environment. Urban planning was chosen because it is a genuine application for CVEs that is useful for the real world (participants can review developments before they are built), and it is compatible with non-naturalistic interface elements (it can have functionality that is not available in the real world, unlike training environments which typically intend to mimic the real world as closely as possible).

Urban planning is an area that would benefit from 3D representations of real environments (e.g. parts of a city) combined with 3D models of proposed developments. Presently, urban planning officials make use of diagrams of proposed developments and photographs of the site to be developed. Unfortunately, it is difficult to understand how the proposed developments will relate to the surrounding area, and many planning applications are deferred for at least one site visit. A 3D representation would provide a realistic representation of the developments that everyone could understand (including members of the public), and it could also provide context by showing the surroundings. A collaborative virtual environment would fit nicely into the public consultation phase of the application process, where neighbours are asked if they would like to support or object to the proposals.

In the evaluation described in Chapters 4 to 6, the participants’ task was to review a residential estate by answering a set of questions, and illustrating their answers with screenshots of the environment (to help prevent ambiguity).

The questions were adapted from UK urban planning guidelines in [19] and [29], see Appendix A. The questions were as follows:

- **Question 1, Permeability:** (a) How many entrance and exit points are there around the estate? What are these for (i.e. cars or pedestrians)? (b) What reduces the speed/volume of traffic? (c) Are there suitable pedestrian routes around the environment? (d) Are the blocks small enough or do you have to walk too far before you reach a choice of direction?

- **Question 2, Character:** (a) Which parts of the environment follow the same pattern/building structure? (b) Find a part of the environment that is not consistent with the layout of the estate. (c) Is this acceptable or should it be changed? (d) Does the estate have character?

- **Question 3, Safety & Security:** (a) Comment on the safety and security of the estate based on your own thoughts, the information in the guidelines and your discussion with other participants. (b) Find examples of where public and private space is clearly distinguished and where it isn't. (c) Discuss which part(s) of the estate you think are least safe. (d) Can you find any blank walls that you think should be overlooked to improve the feeling of safety and help prevent graffiti? (e) Try to suggest some improvements with regard to the safety and security of the estate.

The environment was based on a real estate in Leeds. The estate was chosen after a murder took place which highlighted one way in which the estate's design didn't follow present UK urban planning guidelines. It occurred in a private space that was only partially enclosed—it was not separated from a public footpath that ran along side it, and on the other side of the footpath was a public park. This broke the following guideline:

> 'Clearly defining and enclosing private space at the back of buildings provides for better privacy and security.
>
> - Back yards or inner courtyards that are private or communally shared space are best enclosed by the backs of buildings.
> - The rear gardens of houses are more secure if they back on to other gardens, rather than side roads, service lanes or footpaths.' [29, p. 23].

The incident served as a reminder of the importance of good design. Unfortunately, the pressures for short term financial savings have been known to compromise good design, and mistakes remain for years to come [26]. An annotated map of the estate is shown in Figure 2.2.

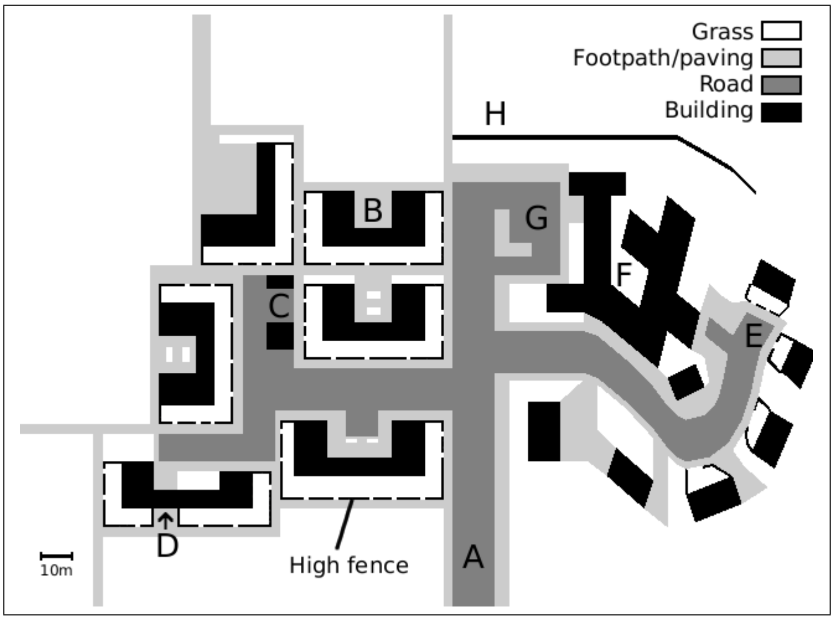


Figure 2.2: A map of the estate. The estate had an entrance road in the middle (point *A*), which acted as a dividing line between two styles of building. On the left-hand side of the entrance road, there were brown-bricked terraced houses, which were mostly horseshoe shapes creating partially enclosed private space (e.g. point *B*). The front gardens were bordered by high fences, and there were six garages in the road (*C*). There was an archway under one of the terraces (*D*). On the right-hand side of the entrance road there were red-bricked bungalows (single story buildings) along the edge of the curved road, with gardens bordered by low brick walls (e.g. *E*). There was a single-story care home for elderly people (*F*), with a car park to the left with space for six cars (*G*), and a hedge-row above it partly separating private land around the care home from public parkland (*H*).

2.7 Summary

This chapter has introduced collaborative interaction in general, and provided some examples of computer supported cooperative work. A taxonomy classified various methods of collaboration, and showed how the work in this thesis fits within that framework (Table 1.1).

Collaborative virtual environments were introduced, with a description of the many ways these can be implemented (display technology, devices and techniques for interaction, software and network architecture). The application areas of CVEs were given, along with specific examples, and consideration was given to potential ethical issues involved. A distinction was made between large- and small-scale space, with examples from previous research, and the extra difficulties involved in large-scale interaction were given along with a definition of what constitutes large-scale.

A discussion followed about group dynamics, the processes by which people form themselves into groups and operate. Methods of forming groups and performing teamwork were analysed, and examples were given from related work. Following this, a background was given on the concept of virtual time, in the context of asynchronous interaction in general. Finally, explanations were given of the evaluation process for two types of groupwork: taskwork and teamwork. The scenario used for experimentation in Chapters 4 to 6 was provided to set the scene.

Chapter 3 details the technical implementation of the CVE system used to carry out the research in later chapters. Chapter 4 explains the development of an initial set of MGDs techniques designed to help groups of people work together as they travel around the environment performing the urban planning task, and it presents an evaluation of these techniques. Chapter 5 identifies two main areas for improvement from the original techniques, and implements and evaluates new functionality to address these issues. Chapter 6 explains the new paradigm for asynchronous interaction known as virtual time, and discusses its implementation and evaluation. Finally, Chapter 7 discusses overall conclusions from both synchronous and asynchronous interaction in CVEs, and outlines research questions for future work.

Chapter 3

Technical Implementation

3.1 Introduction

This chapter describes the custom software application that was written to enable the techniques in later chapters to be implemented and evaluated. Custom software was used to create an 'engine' for the CVE system, on top of which the environment and functionality could be built. This was done to make it easy to have complete control over the system and add new functionality. Alternative approaches would have been to build the environment on top of a commercial game engine, or using a VR engine (e.g. XVR [21]). However, the principles described in this chapter apply to any implementation (e.g. using photographic textures, hypothetical vs. accurate construction, audio capture and processing).

The software was written in C++ using OpenGL and OpenAL. The technical implementation of the CVE system is broken down into four components: map building, collision detection, networking and communication. This creates a conventional collaborative virtual environment. The functionality that was added to this to aid collaboration is described in the subsequent chapters.

3.2 Map building

The term 'map' is from the widely used gaming terminology, which refers to the visual 3D representation of an environment, either real or hypothetical, in which a user can navigate.

A CVE system can contain several maps, e.g. they could each represent different parts of a city, or different urban planning proposals. This section describes the creation of the maps for two environments used in the later research: a training environment used to get participants familiar with the system, and the environment for review in the context of urban planning.

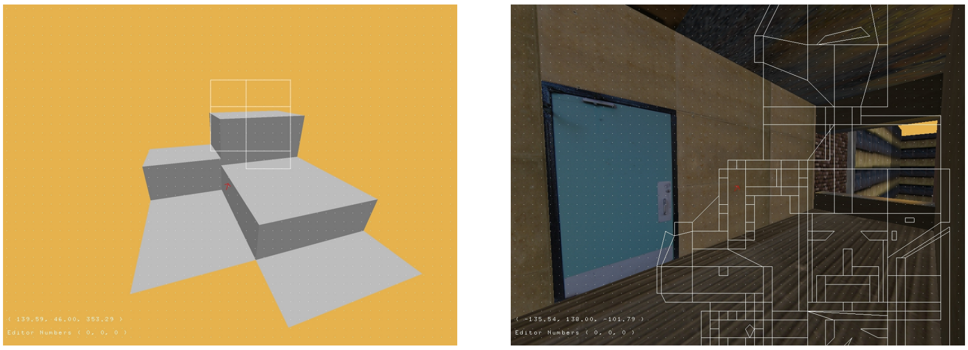
There were three options for map building. (1) A real environment could be accurately reconstructed in the CVE system. (2) A hypothetical environment could be produced. (3) A hybrid of a real environment with hypothetical elements could be produced.

Accurate virtual reconstruction of a real location (option 1) is possible using laser scanning. A laser scanner fires light photons and measures how much time they take to return, to calculate the distance to the nearest surface. The scanner can be rotated and the distance data can be combined with the knowledge of the laser's position and orientation. The resulting data generate a point cloud (a series of data points that lie on surfaces within a direct line of sight from the laser's position). A series of scans can be taken from different perspectives, and polygons can be drawn onto the surfaces found in the resulting point clouds. Finally, photographic textures of each surface can be mapped onto the polygons. This technique is used by [3] and [81].

A hypothetical environment (option 2) could be constructed using map editing software (i.e. the map is hand crafted). This was the option used for the design of the training environment. The main part of the experiments involved participants reviewing the environment in the context of urban planning (i.e. does the environment fit with urban planning guidelines?) In this case there was a danger of creating a hypothetical environment where the arguments for and against the design of the estate were too obvious (they were put in deliberately by the designer). However, very accurate representation of a real environment was not required. Therefore, the hybrid approach (option 3) was taken for the main environment used in the studies. The main environment was *based* on a real estate, followed the pattern and contained the features from the real place, but it was modelled by hand without the use of laser scanning technology. The choice of environment is explained in Section 2.6, and Figure 2.2 contains a 2D plan view of the map, with labels.

A map editor was developed so that the two environments could be drawn using the mouse. Users could draw the outline of a building, and then raise it up out of the ground to a chosen height, and place a roof on top. Figure 3.1(a) shows the user drawing the outline of some buildings, and raising them to different heights. The buildings could then have textures applied to them, by selecting the walls using the mouse. Ceilings could be optionally added, to create indoor spaces, an example of which is shown in Figure 3.1(b).

The training environment was constructed using hand-drawn textures. The main envi-



(a) Some blocks created and raised to varying heights, shown with plain shading.

(b) Indoor areas. The textures in the environment shown were all hand-drawn, with the exception of the door.

Figure 3.1: The map editor.

ronment used photographs from the real location as textures, which provided extra detail to participants.

Figure 3.2 shows some thumbnails of the original photos that were turned into textures for the main environment (photos of walls, fences, garage doors, windows, paving stones, hedges). A total of 170 photos were taken and were made into 70 textures which were used for the final version of the map. Photos of clouds were taken for the texture mapping of the sky. A panoramic photo of the Leeds skyline was taken for the background. Figure 3.3 shows a close-up of some of the textures mapped onto one of the buildings in the environment.

The sky and background photos were wrapped around the scene and were always a fixed distance from the camera, to create the illusion that the buildings in the texture were a large distance away. This technique is known as a skybox (Figure 3.4).

Photographs of people who participated in the studies were taken for their avatars. Photographic avatars meant that everyone had realistic, and recognisable representations of themselves in the environment (the importance of this is outlined in Section 2.3.2.1). Participants had four photos taken of them (front, back, left, right), and an example is shown in Figure 3.5. Each avatar was rendered to participants differently, by choosing a different texture depending on their perspective in the environment. If one was looking at someone face on, then one would see the 'front' photo, and as one moved around to the participant's right-hand side, one would see the texture image change to the 'right' photo.

This was implemented as follows. Each participant's local instance of the CVE application calculated the avatar appearances (avatars were rendered with respect to the participant's viewpoint). The participant's orientation (an angle from $0-360°$) was sub-

Figure 3.2: Thumbnails of some of the original photographs that were made into textures for the 3D environment.

Figure 3.3: The textures used for ‘Burley Willows’, the care home for elderly people.

Figure 3.4: The skybox: the background is permanently wrapped around the scene to provide orientation cues to participants.

tracted from the orientation of each avatar, to calculate the difference. Let d represent the difference, mapped to an angle from $0-360°$. For each avatar:

- $315 < d \lor 0 \leq d \leq 45 \Rightarrow$ the *back* texture was used
- $45 < d \leq 135 \Rightarrow$ *right* texture
- $135 < d \leq 225 \Rightarrow$ *front* texture
- $225 < d \leq 315 \Rightarrow$ *left* texture

Figure 3.6 shows an example of some of the avatars within the environment. The final version of the 3D environment, complete with textures and skybox, is shown in Figure 3.7.

3.3 Collision detection

When a participant moved their avatar around the environment, the CVE system checked for collisions with an object (e.g. buildings, fences), and stopped them from passing through it. To make interaction easier, the system allowed people to 'slide' across surfaces. A basic collision detection algorithm works as follows:

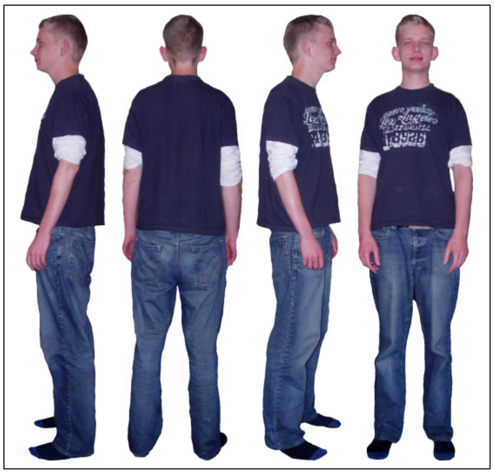


Figure 3.5: The four photos used for the avatar of one participant (left, back, right and front).

Figure 3.6: The photographic avatars of three participants, in front of some terraced houses.

Figure 3.7: The environment used in the urban planning review.

(1) Every frame that a user moves, a ray is calculated from the position of the user's avatar in the previous frame (the ray start) and the attempted new position (which gives the ray direction). This ray is checked to see if it intersects with a collision surface.

(2) If it intersects between the old and attempted new position of the avatar, then there is a collision. A new ray is calculated from the attempted position of the avatar (the ray start) and the normal of the collision surface (the ray direction). The avatar's coordinates are changed to the point where this new ray intersects with the collision surface. This causes the avatar to slide along the collision surface as the user attempts to move against the collision plane (Figure 3.8).

The problem with this method, however, is that it does not take into account the width of the avatar. The avatar would be restricted from moving through surfaces, and would slide along them, but half of the avatar would intersect with the wall (the collision would only be detected when their centre point crossed the surface). This problem was tackled by moving the collision surfaces out along their normal by half the width of the avatar, before checking if they have moved across it (Figure 3.9). This meant that collision took place before the avatar rendering intersected with the wall.

Figure 3.9 highlights a further problem. Moving the collision surfaces out along their normal leaves a gap between the collision surfaces at the edge of objects. The system dealt with this by extending surfaces to cover this gap.

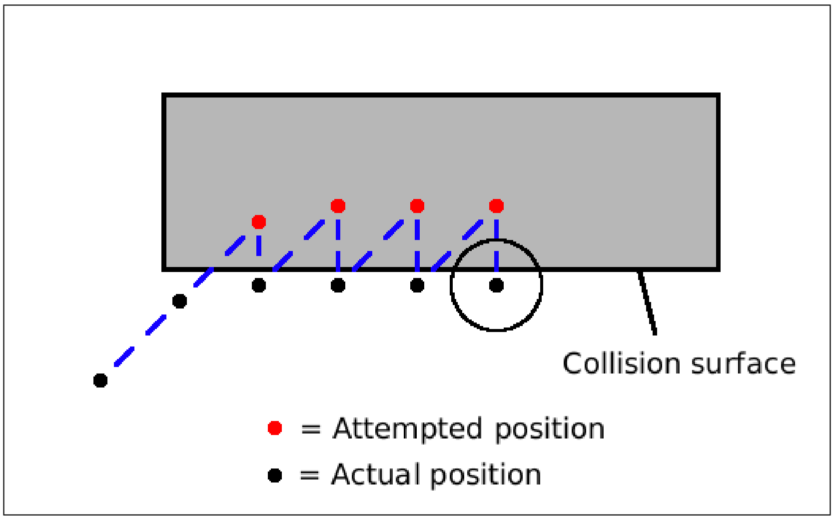


Figure 3.8: The collision detection algorithm. When an avatar moved across a collision surface, it was repositioned by moving it along the normal, to the point of collision. The black and red filled circles represent the centre point of the avatar, and the hollow circle shows the avatar's size.

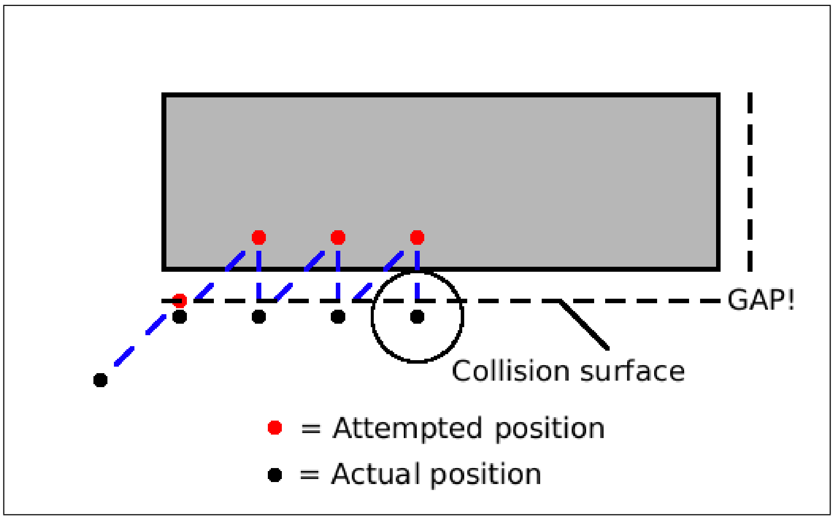


Figure 3.9: The collision surfaces were moved along their normals to account for the width of the avatar. This figure shows that the hollow circle representing the avatar no longer intersects with the visual surface. A gap appeared between the collision surfaces after they had been moved, and the system implementation extended the collision surfaces to account for this.

3.4 Networking

The different possibilities for network architecture and potential problems (e.g. consistency vs. throughput) are outlined in Section 2.2.3. This section describes the networking implementation chosen for the CVE system used in this research.

There are four types of information that need to be sent across the network in a conventional CVE. (1) The position and orientation of participants' avatars, (2) the text communication, (3) the audio communication, (4) system events such as a participant logging into the system or leaving the environment. Functionality introduced in later chapters added other categories to this, such as participants forming/joining/leaving a group, and these were implemented using the same principles as the other four types of information.

The architecture chosen for the CVE system was a client-server model, where the server maintains the shared state of the environment. This avoided conflict between clients (the server had the 'master' state), and a network throughput of 10Hz was used for type 1 data to keep every client updated. Type 2 data was transmitted one character at a time, to reduce the delay between participants starting typing and others receiving the text, and backspace characters were transmitted so that receiving clients could update their representations of the text accordingly (participants could see the typing corrections of others). Type 3 data was transmitted when participants held down their 'talk' key. The talk key operated like a 'push-to-talk' button on a two-way radio. This meant audio data was not being transmitted unless participants had something to say (this saved on network bandwidth). Type 4 data was transmitted when events occurred, such as a participant logging into the system, and this type of data was used to bring new clients up to date with the current shared state of the environment. For example, when a new participant logged on, their client application needed to know how many other participants were connected, where their avatars were positioned, and any text that was currently being displayed in speech bubbles. This information only needed to be transmitted by the server once to each new client—after this only *changes* to the state needed to be transmitted. For example, if no participants were moving, no type 1 data was transmitted. Type 2 data was only sent when someone typed a new character, and type 4 data when someone else logged on or off. The overall state of the environment was not transmitted every frame, because this would take up unnecessary network bandwidth.

A participant's avatar movement on their local machine appeared smooth, updating every frame. If the client positioned the other participants' avatars at the coordinates received from the network, it would give slightly 'jumpy' movement (the movement would be updated at a lower frequency, since movement data was not transmitted every frame to save network bandwidth). CVEs deal with this problem by using dead reckoning algorithms (Section 2.2.3) which predict where objects are going to be at the next frame based on the most recent position and velocity data received from the network. This creates smoother movement, but may mean the prediction is incorrect, and simply placing the object at the correct place when an update is received from the network would cause it to 'jump' to the new location. The CVE system used in the research for this thesis dealt with this problem in a simple way. It interpolated avatar positions and orientations between the last two most recent updates received from the network. This meant that the movement was smooth and no corrections needed to be made. A consequence of this was that avatar positions were slightly out of date, but this was not important in the context of urban planning, and was countered by a relatively high update rate from the server (10Hz for type 1 data, see above).

All information that is sent across a network is represented as a sequence of bits (0s or 1s) [38]. When the data was transmitted by the CVE system, the instance of the application at the receiving end needed to know what the bits represented. The data was put together into units, inspired by the Protocol Data Units (PDUs) from the IEEE standard for Distributed interactive simulation (DIS) [1]. Each unit contained a different type of data (from the types 1–4, above).

A data unit started with a byte containing the ID of the object that the data related to (e.g. if the bit sequence described type 1 information, then this byte would have contained an identification number for the avatar it referred to). This was followed by a byte specifying the type of the data. Each type of data had a specified size. For example, type 1 data was always five floating point numbers: two for orientation (heading and pitch), and three for position (x, y, z). In the case of type 3 data, the audio was broken down into blocks of 0.1 seconds in length, and each audio data unit contained one block (Section 3.5 explains the audio communication in more detail).

The Transmission Control Protocol (TCP/IP) was used to deal with problems of data packets going missing or arriving out of sequence, and with flow control [38]. TCP is a useful protocol for dealing with congestion on the network, but the application developer should be careful not to cause congestion. Type 3 data takes up the most bandwidth, so this was sent using UDP/IP, and the CVE system only transmitted audio data when a participant pressed their 'talk' key, to help reduce network congestion. The following section gives more detail on the audio processing.

The server recorded all the data transmitted to a log file, with timestamps so that sessions could be played back at their original speed. This was used for analysis of the experiments described in the following chapters.

3.5 Communication

It is common for desktop CVEs to allow text communication using the keyboard, e.g. [2] [70] [65]. Many CVEs use audio communication too (i.e. users have headsets), which in some cases has been implemented using a telephone connection (e.g. [93]), and in other studies has been implemented using Voice over Internet Protocol (VoIP). For example, the Robust Audio Tool (RAT) used in the COVEN project uses VoIP [84] [109]. Skype is a VoIP tool which provides an API for developers [100]. The OpenAL API [50] provides audio capture functionality, and developers can transmit the captured data across the network.

For the CVE system being developed, text communication was implemented by transmitting each character as it was pressed, and each character appeared in a speech bubble above the corresponding avatar as the participant typed (Figure 3.10). When participants pressed their ‘enter’ key, the text remained for a further 10 seconds, before being removed. New text could still be added during this time: a new speech bubble would appear, and this would ‘push’ the old one higher up. This way a chain of speech bubbles could be created, and each would expire after 10 seconds of display, reducing the chain from the top down.

Figure 3.10: The text communication above a participant’s avatar.

Audio communication was implemented using OpenAL 1.1. This was chosen because other systems such as RAT and Skype broadcast the sound to all users, and OpenAL allows the developer to deal with transmission and specify different playback parameters for each user. Transmission could be controlled so that only certain users received the audio, and a 3D sound model was used for audio playback. Functionality was added to the system later to support groups, and playback parameters were set differently for group members (they heard communication from their fellow group members without distance attenuation). This is described fully in Chapter 4.

The sound model used was the 'inverse distance clamped' model, a reference distance (refdist) of 30m and a roll-off factor (rolloff) of 6. This means that for distances (dist) of up to 30m the gain was 1.0, between 30 and 85m the gain was defined by the equation $\frac{refdist}{refdist+rolloff(dist-refdist)}$ [50]. This gave a gain of 0.08 at 85m, and beyond this the gain was set to zero.

The stereo channels were used to help participants pinpoint the sound source. If a source was to the right of the listener or central, then the gain of the right stereo channel was kept the same (the gain calculated by the attenuation model). As the source moved to the left of the listener, the right channel gain was reduced from 100% (central) to 0% (directly to the left of the listener), and vice-versa for the left channel. This is calculated by the OpenAL implementation. Further, an icon was placed above a participant's avatar when they spoke, as a visual cue to help others identify who was talking.

The captured audio data was processed in blocks of 0.1 seconds of audio. Each sample was stored using 16 bits (mono) and samples were taken at a frequency of 20kHz. So a 0.1 second block of audio was 32kbit of data (4kB). This amounts to a rate of 320kbit/s. (For comparison, a Fast Ethernet LAN is nominally 100Mbit/s). The blocks of audio data described in the rest of this section are referred to in terms of the amount of time captured, for the sake of clarity.

The OpenAL implementation has an internal ring buffer which stores the audio being captured. The CVE system waited until 0.1 seconds of audio had been written to the internal ring buffer before requesting it. The data was then copied to a buffer in the application memory, ready for processing.

The OpenAL implementation's internal ring buffer size is specified by the application, and needs to be large enough so that new audio data can be received while the data is being copied to the application. Otherwise, some audio data would be lost, explained as follows. Suppose 0.1 seconds of audio data has been stored by the OpenAL implementation. There will be a short time delay before the application detects this and requests the data. During this time more audio data would have been received from the microphone input and been dropped by the OpenAL implementation.

When the CVE was running, some data processing was involved in transmission of audio data (splitting into data units and adding IDs to indicate which person was talking), but the majority took place upon the receiving end. The audio data was received by the server, and put into ring buffers ready to be sent to every client except the originating one. Each client then received the audio data from the server, and put it on a ring buffer for processing. Each data unit was parsed to find the avatar that the audio needed to be associated with. An OpenAL sound source was positioned over the corresponding avatar, and the audio data was queued to this source. The sound source was set to 'play' when enough audio data had been received (0.2 seconds), and new data units were appended to the queue when they were received. The reason for clients buffering the data before starting playback—waiting until they had 0.2 seconds, which is an amount that could be changed by the system designer—was to allow for network congestion. If playback started straight away and the next packet was delayed, then a gap would be heard in the sound, and the repetition of this would cause sound playback to be poor quality. The consequence of the buffering was a small lag (0.2 seconds seemed acceptable), but the main benefit was the increased probability of smooth audio playback.

The processing time was limited. If 10 people were talking at once, then 1 second of data was being received every 0.1 seconds. If this second of data took more than 0.1 seconds to process, then the next second of data would have arrived before the processing was finished, and the amount of data queued for processing would have increased. A large space in memory could be allocated to account for this, but if people carried on talking it would overflow in the end. A recovery system was developed to deal with overflow so that the CVE system would be robust even with large numbers of users.

As specified above, audio data received from the server was stored on a ring buffer for processing (before being queued on a source by the OpenAL implementation). The audio data was dropped if this ring buffer was full. It was important to drop the whole data unit: a policy of 'all or nothing'. Otherwise, some would remain on the ring buffer, and when the next bytes were added to the ring buffer the data unit would be misinterpreted in two ways. First, the system would assume the header bytes of the next data unit were actually audio data from the preceding data unit, and it would try to play them. Second, the system would assume the following bytes were the header bytes of the next data unit, when they were actually audio data, and this would lead to unpredictable behaviour.

The CVE system (including audio communication) was tested and ran successfully on Linux and Windows platforms across the Internet on a home broadband connection (2Mbit/s). It was deployed on a Linux platform across a LAN (100Mbit/s) for the purposes of the evaluations described in the following chapters.

Chapter 4

Mobile Group Dynamics

4.1 Introduction

This chapter describes the implementation and evaluation of techniques called Mobile Group Dynamics (MGDs), which helped groups of people to work together synchronously while they travelled around large-scale collaborative virtual environments. The techniques were evaluated using an urban planning scenario in which one group of participants were provided with MGDs functionality and another ('control') group were not.

4.2 Designing and implementing MGDs

The goal was to develop techniques that helped people work together over an extended period of time, in a large-scale space. This was achieved by designing MGDs to support the *forming* and *performing* stages of Tuckman's model of group activity (Section 2.3). In other words, the MGD techniques were designed to make it easier for groups to *form* in the CVE, and to support their operation as they *performed* the task. The MGD techniques differed from prior work by using a novel 'group graph' metaphor for users to keep track of each other (Section 4.2.1) and an easy mechanism for switching between moving as individuals vs. a group (Section 4.2.2). The techniques are described here, in the context of forming and performing, and Section 4.4.1.4 gives full details of the interface controls.

The design of MGDs was an iterative process. Ideas were sketched out and reviewed by presenting them to the Visualization and Virtual Reality research group at the School of Computing, University of Leeds. The questions and comments from the research group were fed back into the design of the system, and the process was repeated.

4.2.1 Forming

There were four key decisions for the forming of groups outlined in Section 2.3.1. These were method of joining, how members are identified, the structure of the group, and the way the group is represented.

4.2.1.1 Method of joining

Participants in CVEs can form implicit groups by gathering together into circles, in a similar way to how people meet and communicate informally in the real world (Section 2.3.1.1). The MGD techniques were designed to provide functionality for *explicit* groups, and the reason for this was that it allowed groups to continue to function when they were spatially separated in the environment, and functionality was combined with this to enable group members to remain in contact at all times (Section 4.2.2.2).

Explicit groups were implemented by making the server maintain a record of group membership (there was no ambiguity), and the groups were made identifiable to participants in the environment so that they could locate fellow group members and get back together again after periods apart (Section 4.2.1.2). However, the method of joining the group could be both implicit (using spatial positioning) or explicit (using selection), and this is detailed below. The implicit option was implemented so that participants who gathered together into circles would be automatically placed into a group by the server, to encourage people to use the grouping system. The explicit option was provided so that participants did not have to be spatially collocated in the environment to form or join a group—they could select an avatar from a distance.

Forming or joining a group could be done implicitly or explicitly, under one of the following conditions:

- **Implicit:** Moving within 1m of another participant's avatar.
- **Explicit:** Selecting another participant's avatar.

A new group was formed if neither participant was already in a group. The group was joined if one participant was not in a group and the other was. If both participants were in different groups, then the implicit condition had no effect. For the explicit condition, one selection would move the participant out of their current group, and a second selection (or satisfying the implicit condition) was required to move them into the other's group.

4.2.1.2 How members are identified

A prototype of the system represented the groups using a side-bar covering a sixth of the screen width. Each participant in the CVE had an icon representing them, which was displayed in this bar. When two or more participants formed a group, their icons were moved so that they were adjacent, and a box was placed around them to indicate group membership (Figure 4.1). The icons were updated when new participants joined or when a group member left.

Figure 4.1: The prototype system, with groups shown on a side-bar on the left-hand side of the screen.

This prototype system did not represent groups within the 3D environment itself. When a participant saw an avatar, they would have to look up its corresponding icon on the side-bar to determine which group it belonged to, which was relatively time consuming. Feedback on this suggested that it could be much simpler (with fewer cognitive overheads) if something in the 3D environment itself could represent group membership. This was achieved by allocating a unique colour to each group, and it was decided that avatars would have something of that colour to denote membership to that group.

It was also suggested that it would be beneficial to have an easy way of locating fellow group members within the environment, so that collaboration could continue when members were spatially separated. An idea was presented of a separate map screen, which would show a plan view of the environment, and connect group members using lines of their group's colour. In other words, the avatars of each group would form a graph (the avatars being the vertices and the edges denoting group membership), which tracked them as they travelled around the environment. The edges of each graph would be coloured according to the colour of that group, so each graph would have a unique colour. This would mean participants could follow lines from their avatar, and easily locate the whereabouts of fellow group members. Further, it would provide interesting information about the spatial positioning of groups (collocated vs. distributed across the environment, and any members travelling around a separate part of the environment on their own would be easily identifiable by a line that connected them to the 'core' of the group).

However, the idea of a separate map screen would have caused problems of disembodiment: people would be removed from the perspective of their avatar and 'pulled' out of the 3D environment view. Participants could be in the main view of the 3D environment, or in a separate view using the map, but not both simultaneously. Would participants try to navigate using the map alone, and lose the feel for the environment? Would participants miss text communication because they were too busy looking at the map, and would miss the speech bubbles above other's avatars? How do you represent to others that they are looking at the map, and not at what they are doing in the 3D environment?

It was decided to leave out the map at this stage, but the idea of the group graph would still be used. The graph would be shown on the avatars themselves. This was particularly clear in a bird's-eye view, when the groups could be seen as independently coloured graphs, with the names of participants at the nodes. The bird's-eye view of the group graph is shown in Figure 4.2. (The bird's-eye view was provided to participants). This meant that groups were explicitly represented within the environment using colours and by following the lines they could find group members. With the introduction of the group graph, the side-bar was no longer so important, and so it was reduced to a small transparent overlay, using a Head-up display (HUD) metaphor, to save on screen real estate.

Delaunay triangulation was used to determine which edges in the group graph should be drawn, thereby reducing clutter in the environment (Figure 4.3). The consequences of this were that participants didn't necessarily have a line from their avatar to every group member, and the edges changed as participants moved.

4.2.1.3 The structure of the group

The group system was hierarchical, and this worked with explicit selection only. A subgroup was formed if selection occurred and both participants were already in the same

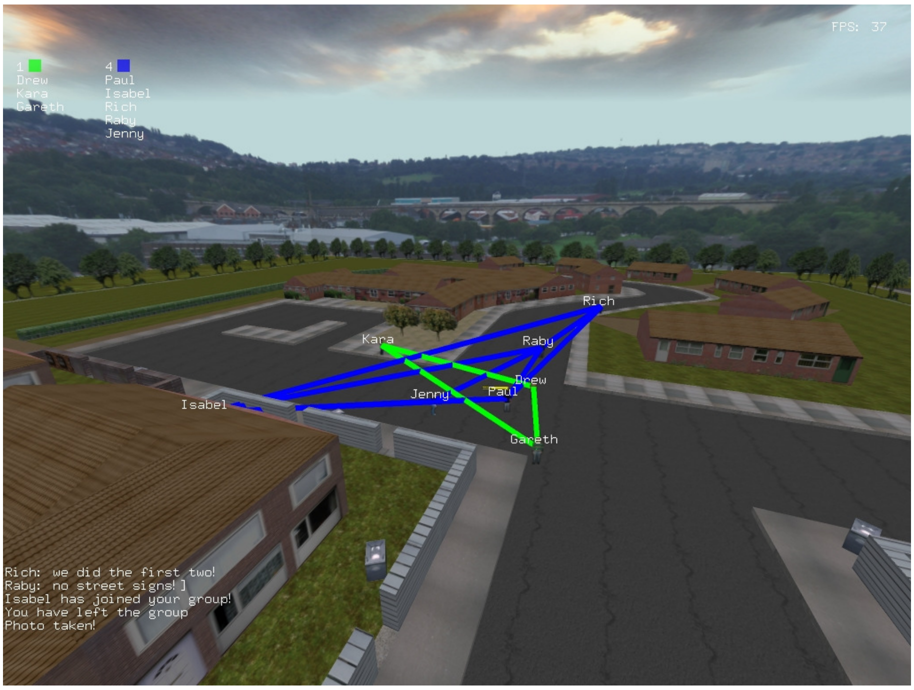
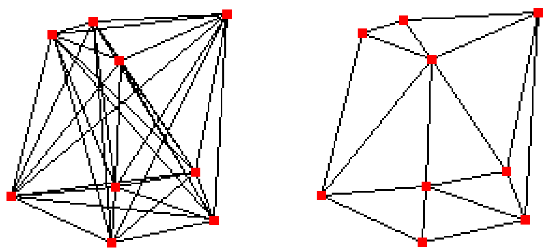


Figure 4.2: A screenshot from MGDs condition, showing two groups from a bird's-eye view. One group is tracked by a green graph and the other by a blue graph. The two graphs have had Delaunay triangulation applied (this is how a participant would see them). Group membership is listed on the HUD at the top left-hand side of the screen.



(a) Without triangulation. (b) With triangulation.

Figure 4.3: An example of the appearance of a graph if all vertices are connected (a complete graph), compared to after Delaunay triangulation has been applied. Delaunay triangulation was used in this context to reduce clutter in the environment, while keeping the graphs connected.

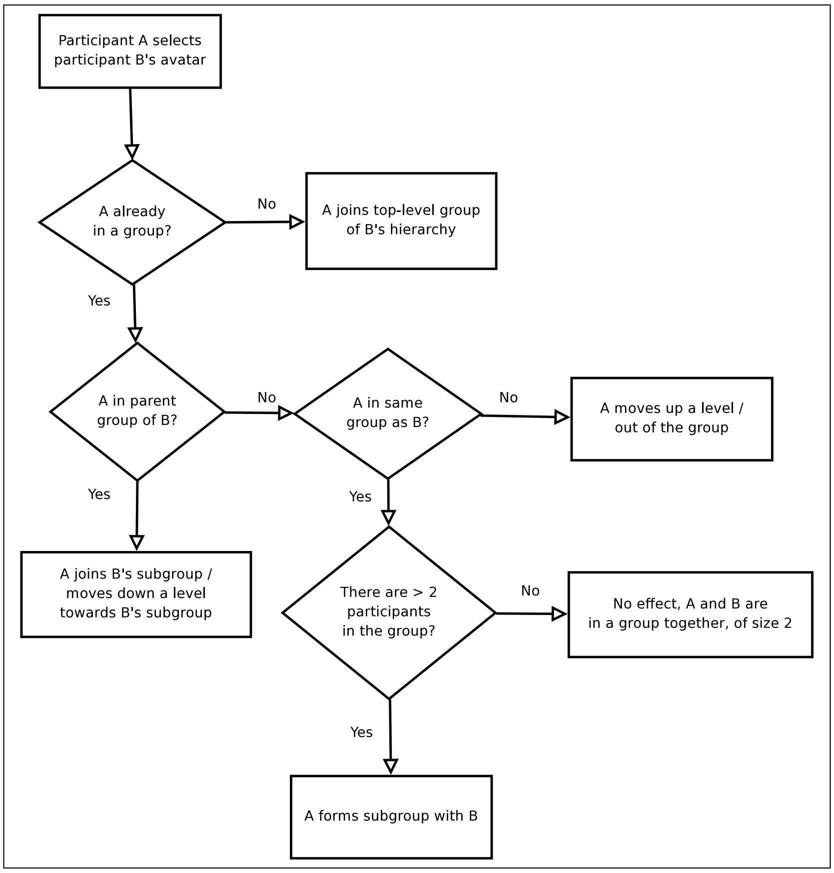


Figure 4.4: A flowchart showing the possible outcomes when participant *A* selects participant *B*. Each selection gradually moves A towards B's group, until they are in a group together.

group. Leaving the group happened one step at a time. First participants would be returned to their parent level of the hierarchy if they were in a subgroup, and they would be removed from their group altogether if they were at the top-level. The grouping system is illustrated with a flowchart in Figure 4.4.

Subgroups were displayed by using a different colour in the group graph, with participants still connected to the graph of their parent group. The parent group's graph was made semi-transparent, so that it could be distinguished from the graphs of the low-level groups.

4.2.1.4 Representation

The way the group is represented refers to different ways the group appears for different participants, and in related work it has primarily been used to reduce rendering requirements and network load (Section 2.3.1.4). Representation was the same for participants in the first study, but perspectives were changed in the next, Chapter 5, where different information and different views were provided for each participant (cf. awareness, multiple views).

4.2.2 Performing

MGD functionality was designed to support groups in the performing of activity. When a group of people work together in a large-scale space in the real world, they communicate and move around the environment (Section 2.3.2). MGDs were designed to facilitate intra-group communication, and to help groups of people move together in the environment, and regroup after periods of separation. The functionality is described below, in the context of the two categories: movement and communication. Related work in these categories is outlined in Sections 2.3.2.1 and 2.3.2.2.

4.2.2.1 Movement

There were two metaphors for moving as a group: the magnetic metaphor and the elastic metaphor. The magnetic metaphor was based on the idea of the CVE system automatically pulling fellow group members together, as if each member generated their own magnetic field. It was implemented as a test, and found that problems occurred depending on the strength of the magnetic pull towards group members. If the pull was too high it was frustrating when one wanted to move apart from the group (movement away was slow), and if it was too low it left one trailing behind when people started to try and move together. If one wanted to stay still it was awkward regardless of the strength of the pull, since other members interfered with one's position. The observations were comparable to the research carried out by Linebarger et al., where they found that users did not like it when external forces interfered with their actions, even if it was the system trying to help [66]. The problem was not in the functionality itself, but in the way the system tried to make changes automatically—the users with manual control found it useful.

This was solved in two ways. First, the functionality was improved: it was noted that to help users move together as a group, the magnetic pull should be towards the mean location of the group, and not the nearest group member (so one is pulled towards the group core, not those who straggle behind). Second, it was made a manual operation. Participants could press a button to activate the magnetic metaphor. This meant it did not interfere with the participant as they travelled around the environment (their position was not affected by the movements of others). Instead, a single button press pulled the participant to the mean location of the group, and the magnetic force was switched off when they arrived close to the mean location, or when they pressed a movement key (i.e.

as soon as they wanted to travel elsewhere they were ‘released’).

The original prototype for the elastic metaphor was designed so that participants could automatically follow a group member. A participant could be automatically carried along with someone, but with the freedom to change orientation and move away from them. The metaphor was that they were attached to the other participant’s avatar by a piece of elastic, so following was easy (automatic) but they had the flexibility to stretch the elastic and move away to take a look at something in the environment.

On paper, the elastic metaphor worked in three steps. (1) Participants selected a group member to automatically follow, (2) participants moved away from them (e.g. to look at a point of interest that they were passing by), and (3) participants could automatically be returned to the location of the person they were following, and continue being carried along with them. It was noticed that the first and third steps of this were very similar, and in order to achieve step 3, a participant could just use step 1 again.

In the final implementation, the system was designed to work as follows. First, step 1 was implemented so that the user could make the selection from any part of the environment—they didn’t have to be close to the avatar they selected. When a participant selected an avatar to follow, they were rapidly moved to their location. Therefore, the need for step 3 was eliminated: step 1 could be repeated instead. Second, the concept of any kind of interfering elastic ‘pull’ was removed when participants walked away to look at something in the environment (users don’t like interference, see above). A participant could perform step 2 simply by pressing a movement key, so manually moving in the environment stopped the automatic movement functionality—they were released in a way that was consistent with the magnetic metaphor. Unfortunately, without the elastic pull back to the person they were following, the elastic metaphor no longer fits. The metaphor is instead simply one of leader/follower, where one participant says ‘follow me!’ and any volunteers are led to another part of the environment. More than one participant can follow the same leader.

In summary, a suite of functionality was provided to assist movement as a group. Participants could automatically follow a group member (leader/follower) or move to the mean location of their group (magnetic), and another benefit of this functionality was that it could be used to rapidly move to a group member’s location. During automatic movement participants still had full control of their orientation, so they could look around while being ‘taken’ somewhere. This meant that they could get an understanding of where they were heading, and continue to look at things in the environment while the automatic movement carried them along. To stop the automatic movement (e.g. to stop following someone), participants simply pressed one of their movement keys.

Collision detection remained enabled during the movement to the mean location of the group because the mean location might have resided in an out of bounds area (e.g. a building). A sliding algorithm smoothly moved participants along walls/fences (Section 3.3).

4.2.2.2 Communication

The communication model in the environment provided 3D audio communication (Section 3.5). This had the benefit of providing participants with a clearer indication of the the location of someone who was talking, which they could work out from the 'direction' that the sound was perceived to be coming from. This was particularly helpful if the avatar associated with the participant that was speaking was out of sight.

MGD functionality was implemented so that the volume level of audio communication was not affected by the distance between people, when they were members of the same group. This helped collaboration continue even when the speaker and listener were far apart in the environment. However, distance attenuation was implemented for inter-group communication (as per Section 3.5), to reduce the overall noise levels.

There were initial ideas to help participants communicate their findings to others in the environment, by enabling view sharing. This would solve the two communication and perspective problems outlined in Section 2.3.2.1. However, this was left out of this first study, so that functionality would be introduced and evaluated in stages. The design and implementation of view sharing, and other new techniques, can be found in Chapter 5.

The rest of this chapter discusses the first study of the participants behaviour in the CVE—a study to see the effects of the new functionality (the MGDs condition) when compared to a conventional environment (the control condition).

4.3 Hypothesis

The MGD techniques were designed to improve synchronous teamwork in CVEs, as detailed in Section 4.2. Teamwork was measured using the methods outlined in Section 2.5.2, and the metrics were the amount and type of communication between participants, the spatial positioning of participants within the VE, and the forming of explicit groups using MGD functionality. In addition, the movements of participants were recorded to a log file by the server, and the CVE software used this to provide an automatic drawing of the participants' paths around the environment, which were used for a qualitative analysis.

It was predicted that there would be more teamwork taking place in the MGDs condition. Participants in the MGDs condition were predicted to communicate more about the task, and to be in closer spatial proximity. It was expected that participants in the control condition would work together to a lesser extent due to the problems inherent in interaction in conventional CVEs (Section 2.3.2).

Participants in the MGDs condition were predicted to collaboratively navigate using the automatic leader/follower mechanisms, and a level 2 qualitative analysis of their paths would determine if teamwork was taking place (i.e. they were deliberately moving together and regrouping vs. their close proximity was incidental — see levels 1 and 2 in Section 2.5.2).

4.4 Experiment

Participants were asked to review a 3D representation of a residential estate that was presented in a CVE system, and complete an urban planning report (Section 2.6). The experiment was carried out in two batches. Participants in the first batch were provided with the MGD functionality that we'd developed to aid collaboration in large-scale VEs (MGDs condition, Section 4.2), whereas in the second batch MGDs were disabled so functionality was typical of current CVEs (a 'control' condition).

4.4.1 Method

The experiment took place in an undergraduate computing laboratory. Each participant used two adjacent computers, one for the CVE, and the other for the write-up of their urban planning report. Participants were spaced out across the laboratory so they could only communicate using audio and text communication from within the environment.

4.4.1.1 Participants

All participants were undergraduate students from the School of Computing. Ten participants were recruited for each run, but two participants for the MGDs condition were unavailable on the day of the experiment. The remaining eight participants (5 men and 3 women) had a mean age of 20.8 ($SD = 2.0$). The ten participants (9 men and 1 woman) in the control condition (MGDs disabled) had a mean age of 22.0 ($SD = 3.5$).

All the participants volunteered for the experiment, gave informed consent and were paid an honorarium for their participation.

4.4.1.2 CVE application

The system allowed multiple participants to connect simultaneously to the environment, be aware of the position and orientation of each other, and communicate using audio and text mediums. The server recorded all activity to a log file (e.g. avatar movements, audio and text communication), with timestamps so sessions could be played back at their original speed, and this playback was used for analysis.

The stereo channels were used to help participants pinpoint the source of audio communication (using a 3D sound model, described in Section 3.5), and an icon was placed above a participant's avatar when they were talking, as a visual cue. Audio volume was attenuated as distance increased between the listener and speaker, to mimic the way verbal communication works in the real world.

Distance attenuation was turned off for communications between members of the same group in the MGDs condition. This helped group members communicate as they travelled to different parts of the estate.

The experiment took place on a Linux platform across a 100 Mbit/s LAN. The CVE system is described fully in Chapter 3. The user interface controls are described in Section 4.4.1.4.

4.4.1.3 Avatar

All participants were represented in the environment with a photographic avatar (using four photos: front, back, left and right, see Figure 4.5). Participants were given an over-the-shoulder perspective, with the option of switching to and from a bird's-eye view. An over-the-shoulder perspective meant that participants could see each other relative to their avatar, and be more aware of how others perceived them [22].

4.4.1.4 User Interface

The participants used desktop workstations, and a two-handed control method, with one hand on the keyboard and the other hand on a 3-button mouse. By holding down appropriate arrow keys a participant could move forward/backward/left/right at 6 m/s, and heading and pitch could be changed by moving the mouse. This is a common gaming control method (e.g. [18]).

The 'Insert' key was used to take screenshots, the 'Home' key to toggle between over-the-shoulder and bird's-eye views, and holding down the 'Page Down' key allowed the participant to use voice communication.

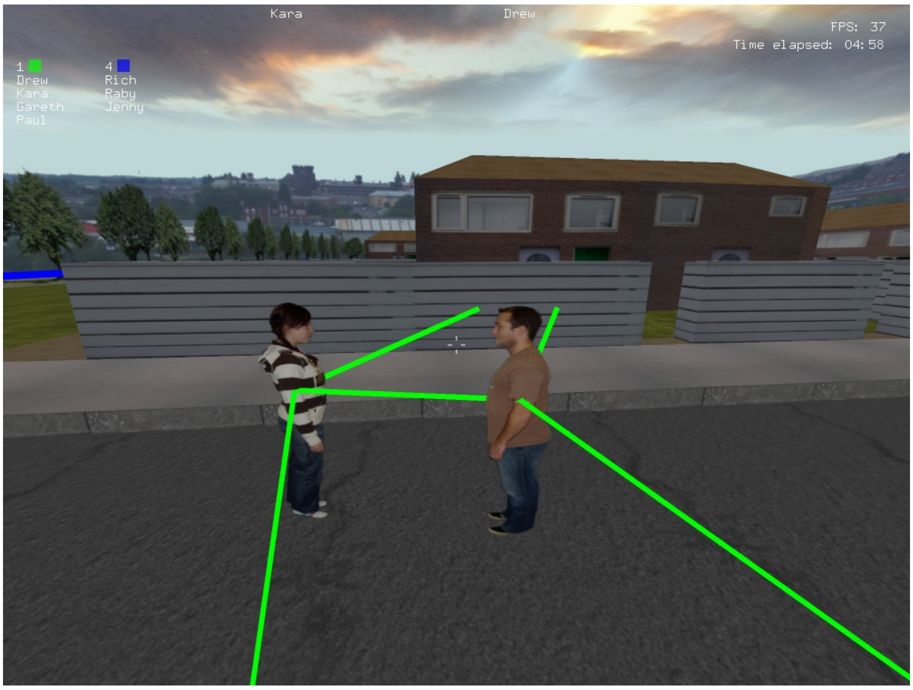


Figure 4.5: The avatars of two participants from the MGDs condition.

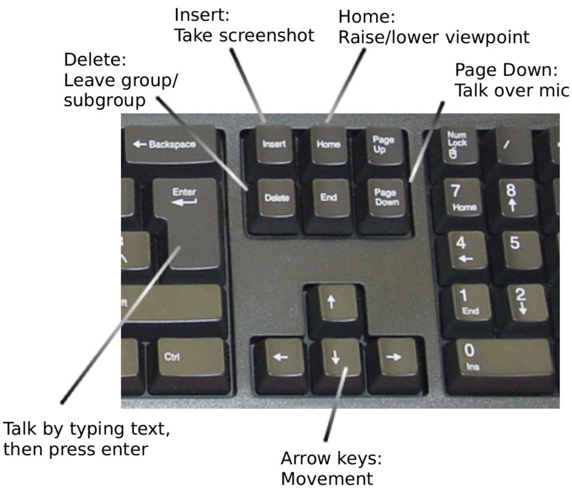
Text communication was achieved by simply typing letters or numbers, which were transmitted the moment each was typed, appearing in a speech bubble above the participant's avatar. The text expired after approximately ten seconds from the moment the enter key was pressed. Each participant was provided with a stereo headset for audio communication. The default recording and playback volumes were automatically set using a shell script.

MGD functionality used three mouse buttons, and the 'Delete' key to move up one level in the group hierarchy. The display had a crosshair in the middle used for selection. Selecting an avatar with the left mouse button formed/joined a group. Selecting the avatar of a fellow group member with the right mouse button rapidly moved to their location and automatically followed them. Pressing the middle mouse button anywhere moved to the mean location of the group.

The keyboard and mouse controls are illustrated in Figure 4.6.

4.4.1.5 Procedure

The experiment used an urban planning scenario, outlined in Section 2.6. Several days before the experiment, each participant attended a ten minute preparation meeting, to have photos taken for their avatar, ask questions about the experiment, and read an introductory sheet containing extracts from government urban planning guidelines (Appendix A).



(a) Keyboard controls for both conditions, with the exception of 'Delete' which was not used in the control condition.

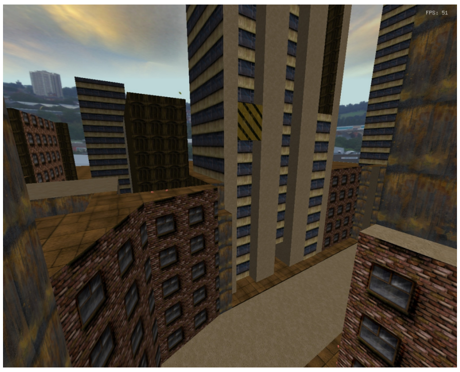
(b) Three-button mouse controls for MGD functionality. Moving the mouse controlled heading and pitch.

Figure 4.6: The user interface controls. The navigation control method was two-handed: participants placed one hand on the arrow keys for movement and the other hand on the mouse to change heading and pitch.

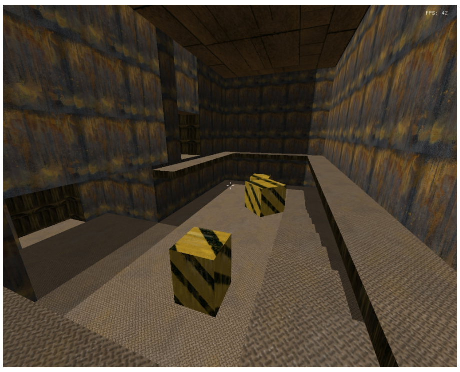
The experiment itself lasted one hour. At the start of the experiment, each participant was provided with three information sheets: another copy of the introductory sheet, instructions that described the CVE's interface (Appendix B), and a schedule for the experiment. They were also provided with an electronic copy of an urban planning report. The questions contained in the report were adapted from urban planning guidelines in [19] and [29], and are listed in Section 2.6.

The first 15 minutes of the experiment were used for training. Participants were instructed to experiment with all the controls available to them, with the experimenter and assistant on hand to clarify anything if necessary. Participants logged into a training environment. This contained a 3D representation of a city, of which an area of approximately 75x75m could be explored. There was a main road area, surrounded by large tower-blocks, with small alley ways around the back of them. Two of the tower-blocks could be entered, one from the road, and the other by descending some steps and going under the road in a subway. There was a lift up to the top of one of the blocks. Two screenshots from the training environment are shown in Figures 4.7(a) and 4.7(b).

The next 35 minutes were allocated for the main task—the review of the residential estate (Figure 2.2). Participants logged into the main environment and travelled around the estate to answer the questions and complete their urban planning report. If a participant came across something relevant to the report, they could take a screenshot of it. The



(a) Outside, looking across at one of the tower blocks.



(b) A storage room inside one of the buildings.

Figure 4.7: The training environment.

screenshot would simply capture what they were looking at, in the same view that the participant had (i.e. over-the-shoulder, or bird's-eye).

The participants received verbal warnings when there were 10 minutes and 5 minutes remaining on the main task, to encourage them to finish writing up the report. The final 10 minutes were allocated to submitting the report, filling in a questionnaire, and receiving payment.

4.5 Results

The data collected can be divided into two categories, taskwork and teamwork—'the work of working together' [7] (Section 2.5). The sources of data were the participants' urban planning report sheets, the questionnaires and the server's recording of everything that took place in the environment (text and audio communication, movement, and the makeup of the teams). The report sheets provided data about the taskwork, and the questionnaires and server's recording provided data about the teamwork.

The server's recording was in the form of a log file. It could be played back, either forwards or backwards (rewinding) at various speeds, and with the ability to move the viewpoint around the environment to view the playback at any position or orientation.

Statistical analyses were performed using independent samples t-tests to compare participants who had been provided with the MGD functionality with those who had not.

4.5.1 Taskwork

The reports were marked like an exam, according to a mark scheme with example answers.

The mean marks were 16.9 out of 24 ($SD = 5.1$) for the MGDs condition, and 17.3 ($SD = 4.0$) for the control condition. An independent samples t-test showed there was not a significant difference in the taskwork scores of the two groups of participants, $t(16) = 0.20, p > .05$.

The task itself was only of modest difficulty, so it was to be expected that performance would not differ between the two conditions. However, the primary interest lay in how MGDs affected the way in which participants tackled the task.

4.5.2 Teamwork

The analysis of teamwork consisted of a combination of two methods based on those in [96]. The first method was quantitative, in which the communication and spatial positioning between participants were analysed, and the results for the MGDs and control conditions were compared. The second method was qualitative, an 'analysis of interaction fragments' [96, p. 661], in which the paths of the core participants in the MGDs condition were analysed to draw out patterns of interaction.

4.5.2.1 Quantitative analysis

For the MGDs condition, each explicit group of participants was given a unique colour. This 'team' colour remained the same despite changes in the combinations of participants who belonged to that team. The teams are shown in Figure 4.8. The participants are shown on the y axis, and given a colour depending on which team they belong to at each point in time, where time is shown on the x axis. The time of zero represents the time that the server was started. Teams were formed from scratch five times, four times implicitly (pairs of participants walked within 1m of each other) and once explicitly (one participant selected another). The chart shows that for the majority of the experiment there were two teams, one blue and the other green, with participants occasionally switching from one to the other.

The data about participants' movements through the environment were used to calculate how far each participant was from their nearest neighbour every second during the experiment. This was then used to determine the percentage of time participants spent separated by given distances from the other participants (see Figure 4.9). These data show

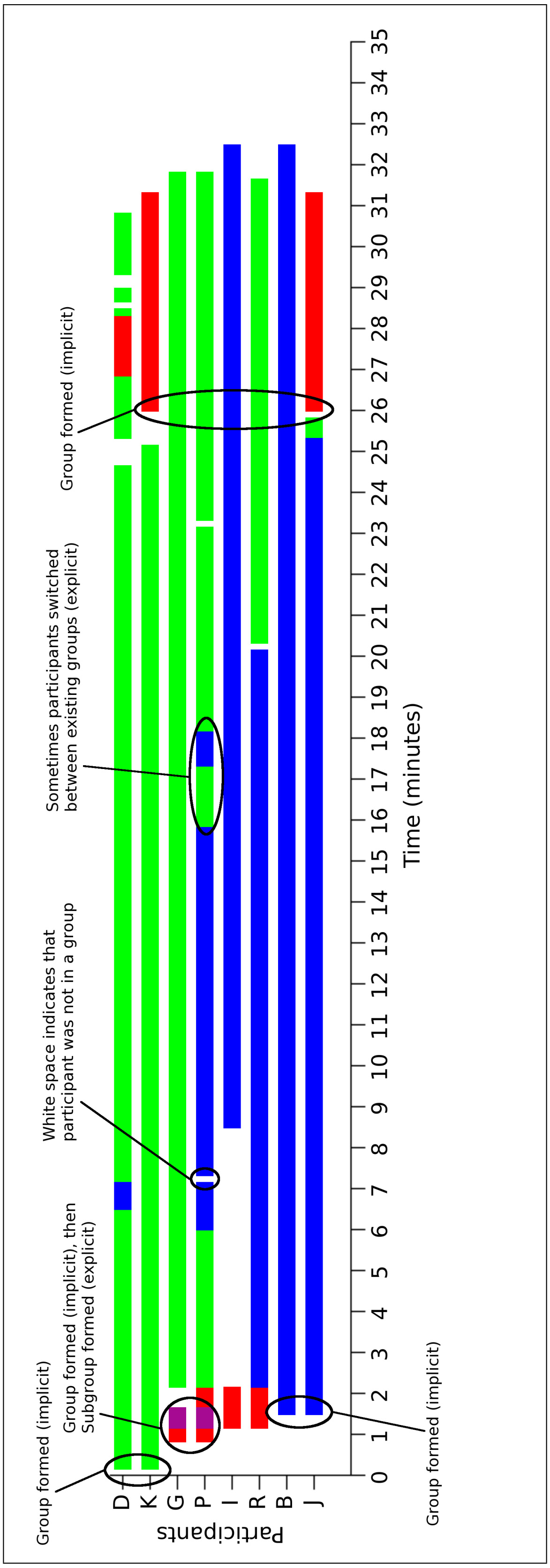


Figure 4.8: A chart showing which team the MGD participants were in over time. Each team is shown in a different colour.

that participants spent nearly twice as much time within 10m of others when MGDs were provided.

The mean distance to the nearest neighbour was calculated for each participant in both conditions. The overall means were 19.7m for the MGDs condition ($SD = 4.2$) and 25.4m for the control condition ($SD = 3.8$). An independent samples t-test showed that there was a significant difference in the distances to the nearest neighbour for the two conditions, $t(16) = 3.05, p < .01$.

The questionnaire was used to gather data on the use of MGDs. In particular, the automatic following mechanism could be used to rapidly move to a group member's location. Six out of the eight participants said they used the functionality in this way.

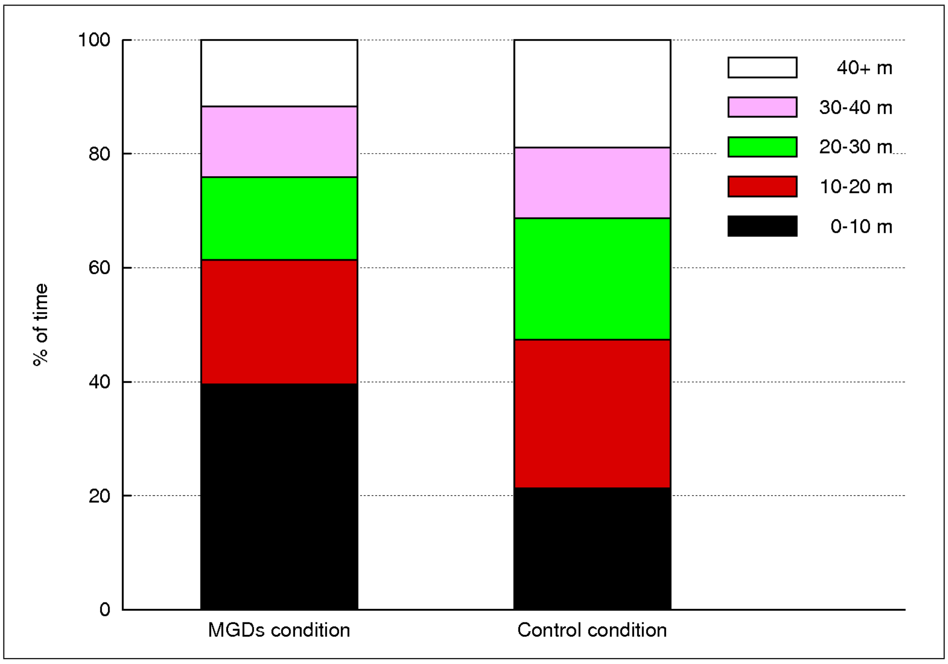


Figure 4.9: Proportion of time participants spent within given distance of their nearest neighbour.

For each batch of participants (the MGDs and control conditions), participant's spoken and text communication was transcripted and analysed using a communication coding approach [15] to classify each utterance as one of the following:

(a) **Greetings** (e.g. 'Hello *R*!', 'How you doing?')

(b) **Functionality** – communication regarding the system and the groups (e.g. '*D* are you following me?', 'Press home to get a better view', 'Can everyone hear me even though we're in different groups?')

(c) **Environment** – discussion about the 3D world, but not in relation to the task (e.g. 'So is this meant to be an actual part of Leeds?', 'There's Leeds city council bins')

(d) **Task related** (e.g. 'What do you think reduces the speed round here?', 'I've found a bit of the estate that doesn't really match the rest')

(e) **Idle chat** (e.g. '*D* I can actually read what's on your T-shirt!')

Overall there were 133 utterances in the MGDs condition (mean 16.6, $SD = 14.1$), of which 40 were text-based and 93 were spoken. The utterances occurred in 22 blocks of conversation and in 15 of these, all the speakers were in the same team. There were 18 utterances in the control condition (mean 1.8, $SD = 1.5$), of which 16 were text-based and 2 were spoken. These utterances occurred in 3 blocks of conversation.

Levene's test showed the variances were significantly different, $F(1,16) = 21.1, p < .001$, and therefore a non-parametric test (Mann-Whitney) was chosen as most suitable for comparing the data from these two conditions. The Mann-Whitney test showed there were significantly more utterances in the MGDs condition ($Mdn = 15.5$) than the control condition ($Mdn = 2.0$), $U = 10.5, p = .006$.

There were 88 task related utterances in the MGDs condition, (mean 11.0, $SD = 9.7$), and 4 task related utterances in the control condition, (mean 0.4, $SD = 0.5$). Levene's test showed the variances were significantly different, $F(1,16) = 17.0, p = .001$, and a Mann-Whitney test showed there were significantly more task related utterances in the MGDs condition ($Mdn = 12.0$) than the control condition ($Mdn = .0$), $U = 9.0, p = .004$.

These data show that there was much more communication in the MGDs condition, and most of it was task-related (see Figure 4.10).

4.5.2.2 Paths during teamwork

In the MGDs condition, the most persistent combination of team members was *D*, *K* and *G*, in the green team, and *P*, *I*, *R*, *B* and *J*, in the blue team (see Figure 4.8).

D and *K* were identified as the core members of the green team because they communicated the most. *D* spoke 29 utterances, *K* spoke 22 but *G* (the third member) only spoke 12 utterances.

R and *B* were identified as the core members of the blue team. *R* spoke 41 utterances and *B* spoke 19, which was far greater than the other members *P*, *I* and *J* who spoke 5, 0 and 5 utterances respectively.

The paths of these core participants from the MGDs condition were analysed in detail and showed that they sometimes moved together around the environment answering a

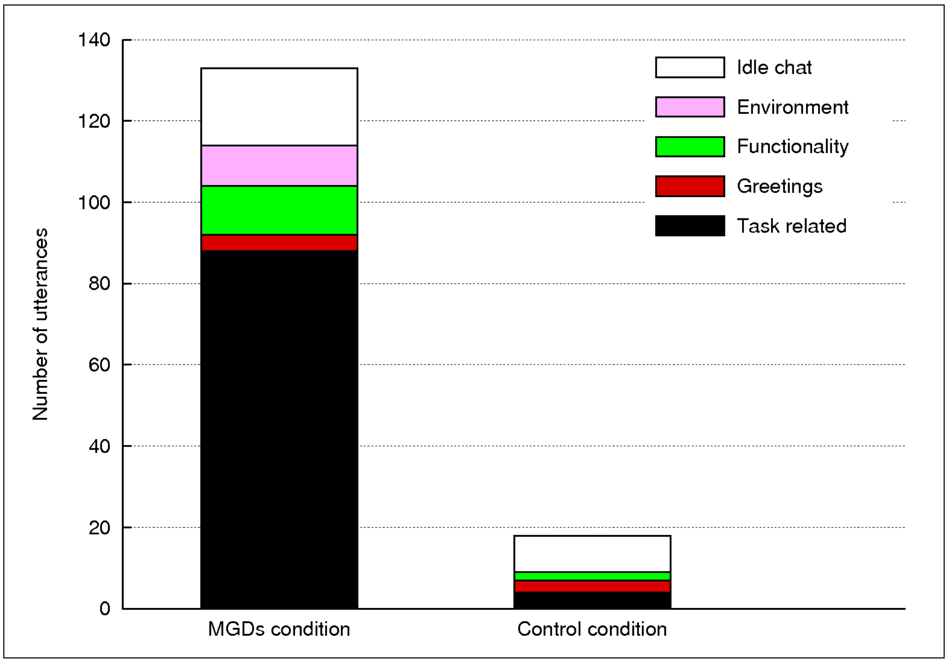


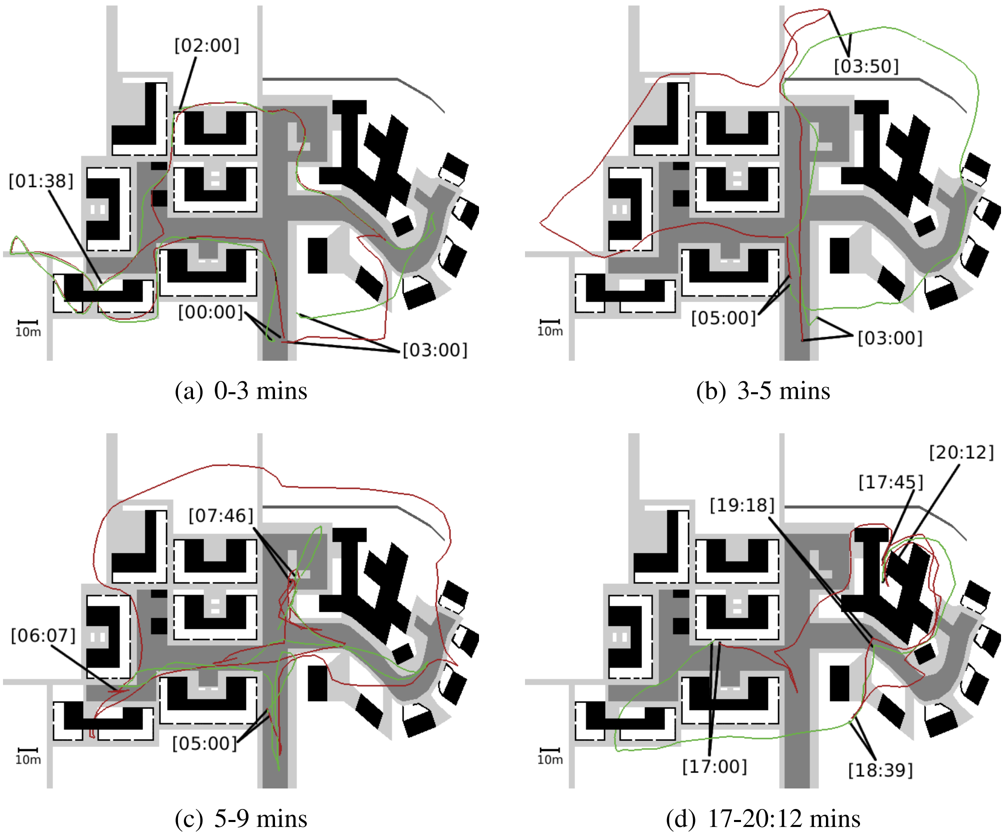
Figure 4.10: The number of utterances in each communication category for the two conditions.

question, and on other occasions split up to explore their surroundings, and then regrouped to discuss their findings. By contrast, participants in the control condition communicated far less and spent little time in close proximity (see section 4.5.2.1).

The following paths and communication from the green team illustrate the types of behaviour that occurred when MGDs were provided. Figure 4.11(a): The two core members of the green team started at the entrance to the estate (shown with a timestamp [00:00] in the diagram), navigated around the environment together in a clockwise direction, and returned to the starting point. *D* was following *K* using the automatic following MGD functionality. Their conversation was based on the functionality of the system (the leader/follower mechanism), and the real world location of the virtual environment. The points at which the conversation took place are shown by timestamps on the diagrams and in the extracts below.

```
[01:38] K: D are you following me?
[01:42] D: I am, yes!
[01:45] K: Wicked!
[01:50] R: I think I can see my house from here!
[02:00] D: So is this meant to be an actual part of Leeds?
```

Figure 4.11(b): The core members returned to their starting point [03:00]. They *split up* [03:50] and navigated one side of the estate each, until they regrouped again in the middle [05:00].



(a) 0-3 mins

(b) 3-5 mins

(c) 5-9 mins

(d) 17-20:12 mins

Figure 4.11: Paths of the core members *K* and *D* from the green team. *K* and *D* are represented by green and red lines respectively.

Figure 4.11(c): The core members *split up* again, *D* navigated the perimeter of the environment and *K* stuck to the roads. *K* met the two core participants *R* and *B* from the blue team and joined in their conversation [06:07].

```
[05:47] R: What do you reckon stops the volume of traffic?
[05:57] B: I don't know
[06:03] R: Could it be that it's so windy?
[06:07] K: Dead ends as...
[06:11] R: Was that...  was that J?
[06:14] K: No that was K!
```

The two core members of the green team *regrouped* at time [07:46]. One of the core members of the blue team, *R*, was with them and joined in their conversation. *R* reported the findings from the blue team.

```
[07:46]/Text D: Hadn't we better start answering some of the questions?
[08:07]/Text K: i already ahve
[08:10]/Text K: haahhha
[08:18]/Text G: probably
[08:36]/Text R: we did the first two!
[08:38]/Text G: how many exits are there?
[08:46] D: I've only found one.
[08:49] R: One what?
[08:52] D: One exit.
[08:54] R: We found four pedestrian and one for cars.
[09:05] R: It's a small world.
[09:08] D: Too true.
```

The time from [09:08] to [17:00] has been omitted because there was little communication between the core members of the green team throughout this time (two utterances from *D* and one from *K*).

Figure 4.11(d): The two core members of the green team *split up*, *D* found something of interest [17:45], they *regrouped* [18:39], *D* showed the rest of the team the point of interest from a distance using the bird's-eye view [19:18].

```
[18:39] D: I've found a bit of the estate that doesn't really match the rest.

[18:42] K: Yeah.  So have I.
[18:46] D: What have you put for that?
[18:49] K: One of the two level houses has got a different colour wall to the
[K is referring to the brown fence around the terraced houses (determined from
K's report)]
[18:55] D: Oh, is that it?
[18:57] K: Yeah.  Why?  What have you got?
[19:00] D: If you press 'Home' and follow me I'll show you.
[19:03] K: OK.
```

D led *K* (and *G* who was listening in) to the large building, and stopped by the side of it to talk [19:18]. (Pressing the 'Home' key toggled bird's-eye view).

```
[19:18] D: If you all look to my left now, and have a look with 'Home'...
[19:23] K: OK.
[19:24] D: It's laid out a completely different way and there's a dead end in
the middle.
[19:33]/Text G: ah yeah i see
[19:34] K: Oh yeah!
```

K and *G* then followed *D* to the point of interest.

```
[20:12] K: I see what you mean.
```

4.6 Discussion

The goal was to develop techniques for Mobile Group Dynamics that helped people work together over an extended period of time, in a large-scale space. MGDs had a neutral effect on task performance (the task was achievable by oneself) but did produce fundamental changes in the way participants went about performing the task and the quantity of teamwork that took place. In particular, this was shown by the amount of time that participants spent near each other, the way they continued to collaborate after periods of separation, and the amount of communication that took place.

Participants in the MGDs condition spent much more time in close proximity (within 10m of their nearest neighbour for 40% of the experiment) than participants in the control condition (21%), and two aspects of MGDs contributed to this. Firstly, participants could easily identify fellow group members because lines between group members indicated the location of others and each group was given a unique colour (see soccer team analogy in Section 2.3.1.2). Secondly, the automatic following functionality helped people remain together while they travelled, and also provided an easy way of regrouping with one's fellow members (75% of the MGDs participants used the functionality in this way).

It is suggested that 'cognitive ease' as well as functionality affects group behaviour in CVEs [8], and this may explain why MGDs were so successful at helping participants collaborate over an extended time that included periods of separation. Firstly, allowing groups to form automatically via spatial proximity minimised the effort involved of initially forming a collaboration with other participants (80% of groups were formed in this way). Secondly, the explicit indication of who was in each team (see above) and the fact that audio communication within a group was not attenuated by distance meant that participants did not lose contact if they wandered away from their fellow group members. Thirdly, leaving or switching groups had to be done explicitly and, therefore, was effortful.

There were over seven times the number of utterances in the MGDs condition, compared to the control condition. The low number of utterances in the control condition could lead to speculation that the participants in the control condition were not willing to talk to each other, e.g. maybe there was conflict amongst the mutual acquaintances, maybe they were not comfortable talking to each other, or perhaps they were experiencing technical difficulties. However, further investigation showed this was not the case. The conversation in the training environment reported a total of 116 utterances in the control condition, with 23 of these being of the greetings category, 46 discussion of functionality, 3 environment and 44 idle chat. For comparison, participants in the MGDs condition made 57 utterances in their training session, 17 of which were greetings, 3 functionality, 3 environment, and 34 idle chat. One participant did have faulty headphones in the control condition training session (which explained the large discussion of functionality as people asked each other if their microphones were working) but spare headsets were on hand and the faulty one was replaced during training and before the actual urban planning review session began. A qualitative analysis of the conversation in the training session for the control condition showed that participants *were* comfortable talking together, as they laughed, joked and tried to sell tickets for the university summer ball, illustrated by the following extract from the conversation transcript:

```
[14:30] A: So tell me, C, tell the lovely people, how much is the summer ball
ticket?
[14:35] C: A, radio personality of the year, 2007.
[14:39] A: That's right, great babe.
[14:41] C: Smashing.
```

This large increase in communication was the result of the suite of techniques as a whole. It could be argued that the very presence of MGDs would have given participants an idea of how to work together effectively [66] and, with 66% of the conversation being task related, this was representative of the extra teamwork taking place.

Finally, although participants could communicate with group members wherever they were in the environment, they still preferred to spatially *regroup* to discuss their findings. When there was a point of interest, it seemed important for everyone to see it from the same viewpoint and get a shared understanding of it (see the dialogue in Section 4.5.2.2, for Figure 4.11(d)). This issue is addressed in the following chapter by improving awareness of who can hear you and who is talking, allowing rapid movement to another location by teleporting, and providing multiple views so participants can see what their group members are looking at.

Chapter 5

Teleporting, Awareness and Multiple Views

5.1 Introduction

Techniques called Mobile Group Dynamics (MGDs) were developed and evaluated (Chapter 4), and they helped groups of people work together while they travelled around large-scale collaborative virtual environments (CVEs). Compared to a conventional CVE, these techniques led to a seven-fold increase in the amount of communication that took place between participants, and participants with MGDs spent nearly twice as much time in close proximity (within 10m of their nearest neighbour). 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 communication between group members). This meant that unnecessary amounts of time were spent travelling 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 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 [79] [39]. Three types of functionality were implemented: teleporting, awareness and multiple views, and the functionality and rationale is outlined in the following sections.

5.1.1 Teleporting

'Walking' is time consuming, so teleporting functionality was added. 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 [86]. The teleporting algorithm took its inspiration from [69], with the addition of gradual acceleration as well as deceleration, and to avoid problems caused by travelling 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 by clicking on the appropriate place in the VE scene. If participants had multiple views functionality (described below, Section 5.1.3), they could click on a thumbnail view to teleport to be next to that person.

5.1.2 Awareness

The hypothesis was that participants collocated to communicate with group members because, given that the CVE used directional sound, participants assumed that the audio was also distance attenuated. In fact, distance attenuation was disabled for intra-group communication, as was explicitly stated in participants' verbal and written instructions. To overcome this, 'awareness' functionality was developed that provided visual feedback about who was receiving audio at a given moment in time, and who was speaking.

The awareness functionality 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 recognised (Figure 5.2). 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.

Figure 5.1: The basic MGDs condition from Chapter 4, shown in the over-the-shoulder perspective. The group graph was used to identify groups.

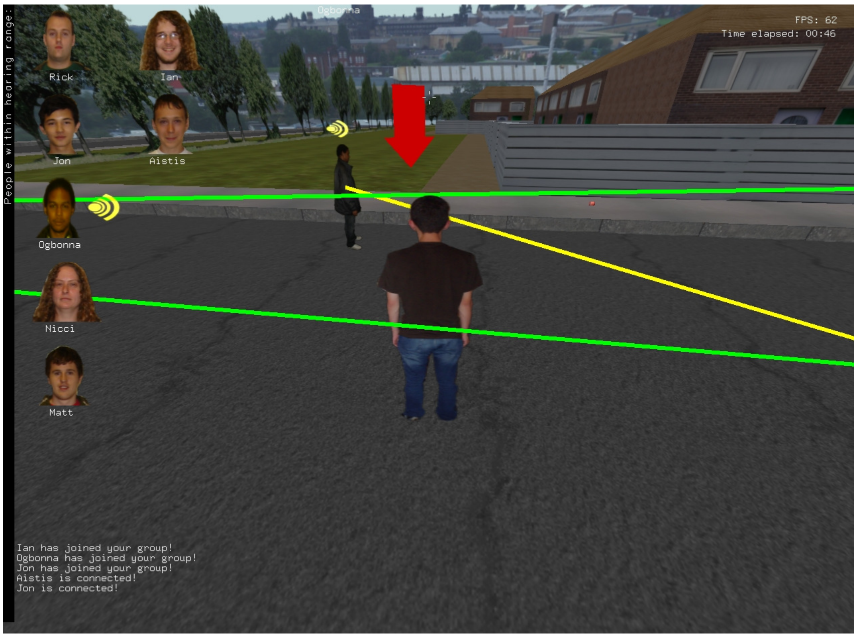


Figure 5.2: The teleport condition, shown in the over-the-shoulder perspective. The red teleporting arrow can be seen above the centre of the screenshot. This condition added awareness functionality, where photographs of participants within hearing range were displayed on the left-hand side of the screen using a HUD metaphor. The speech icon can be seen above the avatar of the participant who was talking, and also next to the participant's photograph on the HUD, useful for when their avatar was out of sight.

5.1.3 Multiple Views

Participants in CVEs often find it difficult to understand what others are looking at [51]. To overcome this, an idea was developed to allow participants to see the viewpoints of fellow group members.

Figure 5.3: The prototype system, with the tabs for switching views across the top of the screen.

There are advantages of both independent and shared (What You See Is What I See, WYSIWIS) perspectives (Section 2.3.2.1). For example, two participants with a shared perspective can understand what each other is referring to, and participants with an independent perspective can explore the environment without interfering with anyone else. The CVE design from Chapter 4 provided participants with their own independent view of the world, for both the control and MGDs conditions. This was extended by providing participants with multiple views.

The original design for multiple views used the tab metaphor (Figure 5.3), which is commonly used in graphical user interfaces (GUIs). The tab metaphor would have allowed the switching from one view (a participant's own view) to another (the view of a fellow group member), much like the tab system used in modern web browsers. However, this would have led to the problem of 'disembodiment', where the avatar's position and orientation would no longer represent what the participant was seeing, and the participant may miss text conversation if they are looking in the wrong window (this problem is introduced in Section 4.2.1.2). To solve this problem, the multiple views were all displayed on the same screen, with no need to switch from one to the next. The participants' own main viewports were augmented with live (real-time) thumbnail views of their fellow group members (Figure 5.4). If a participant wanted to take a closer look at something nearby another user, they could click on their thumbnail view to teleport to that location (Section 5.1.1), and look around from that position. This means that participants' perspectives were shared with other users but controlled independently (other users could not interfere).

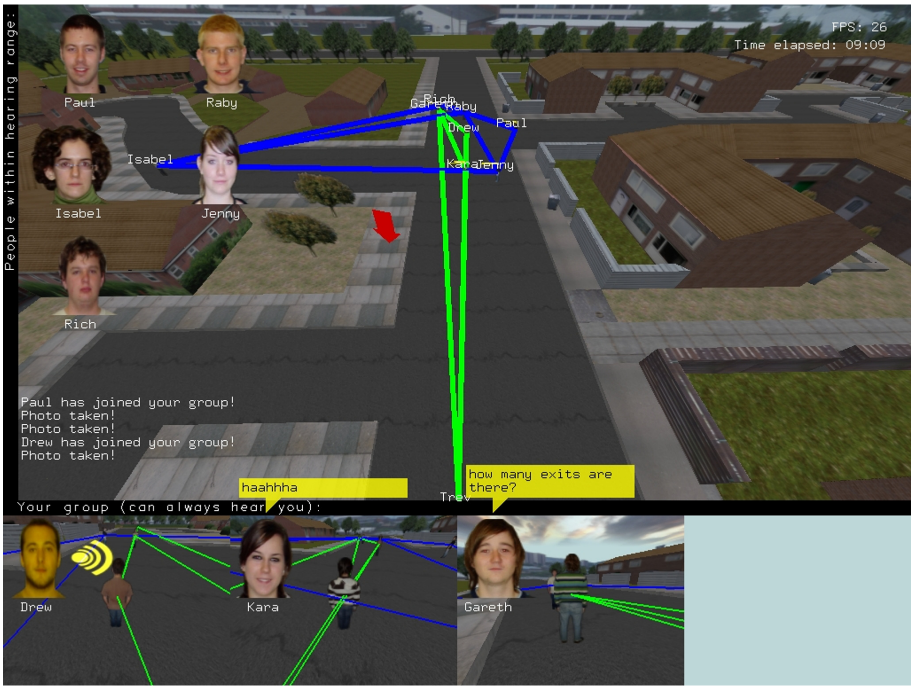


Figure 5.4: The multiple views condition, shown in the bird's-eye view. The group graph, teleporting arrow, and faces on the HUD (from the previous conditions) can be seen here, and this condition added thumbnail views of fellow group members across the bottom of the screen. Note that the text communication was displayed above the thumbnail views, so that it could be read even if the associated avatar was a large distance away. The speech icon was also shown on the thumbnail view to indicate that they were using audio communication.

5.1.4 Hypothesis

The hypothesis was that the teleporting, awareness and multiple views functionality would improve teamwork by tackling the shortcomings found in the original MGDs study (Chapter 4). In the MGDs condition, participants collocated to communicated (which was unnecessary given that MGDs allowed participants in the same group to communicate over an infinite distance), and this was specifically tackled using the awareness functionality, which was designed to provide awareness of this basic MGD functionality (that allowed intra-group communication) by making explicit the participants who could hear you at any given moment.

Further, the qualitative analysis of the MGDs condition showed that understanding another's perspective (e.g. a point of interest they were referring to) was difficult and time consuming, and related to the well known difficulties from previous research — problems 1 and 2 in Section 2.3.2.1. The multiple views functionality was designed to specifically tackle these perspective problems.

The server recorded everything that took place in the environment to a log file (e.g. audio communication, participants movements, usage of teleporting). Most metrics used for evaluation were calculated automatically from this log, and in addition the conversation was transcripted by hand. The metrics were the amount and type of communication between participants, the spatial positioning between participants while communicating, the usage of teleporting and the movements of participants for the qualitative analysis.

It was predicted that communication would increase (an indication of more teamwork taking place), and the distance between group members while communicating would increase (group members would be made 'aware' of the MGDs functionality provided, and not spend unnecessary amounts of time collocating to communicate). It was difficult to predict the affect that teleporting would have on teamwork, other than to say it would increase the efficiency of navigation around the environment and to points of interest. A qualitative analysis of teleporting was required for further investigation.

5.2 Experiment

The experiment used the urban planning scenario outlined in Section 2.6. This was the same scenario as the one used for the experiment in Chapter 4. Participants were asked to use the CVE system (detailed in Chapter 3) to review the design of the housing estate. An annotated map of the estate is shown in Figure 2.2. Participants were run in two batches. In the first of these (the teleporting condition), participants had all the basic MGD functionality from Chapter 4, and new MGD functionality to provide awareness of who was talking, who was within hearing range and teleporting. Participants in the second batch were provided with multiple views (the multiple views condition), in addition to all the functionality that was provided to the other batch of participants.

5.2.1 Method

The experiment took place in an undergraduate computing laboratory. Each participant was provided with a headset, and they were spread out across the laboratory so they could only communicate using audio and text communication from within the environment. Participants used two adjacent computers, one for the CVE and the other for the urban planning report write-up. The CVE application, environment and experimental procedure were the same as in Chapter 4.

5.2.1.1 Participants

All participants were undergraduate students from the School of Computing, who had not taken part in the previous study. Eight participants were recruited for each run, but one participant in the teleporting condition was unavailable on the day of the experiment. The remaining seven participants in the teleporting condition (6 men and 1 woman) had a mean age of 21.7 ($SD = 5.2$). The eight participants in the multiple views condition (5 men and 3 women) had a mean age of 21.8 ($SD = 4.1$).

All the participants volunteered for the experiment, gave informed consent and were paid an honorarium for their participation.

5.2.1.2 CVE application

The software application and 3D sound model are described in Chapter 3.

Distance attenuation was turned off for communications between members of the same group. This was clarified by displaying photographs of the faces of participants who would receive any transmitted audio (the 'awareness' functionality). These faces were displayed on the HUD, and were added and removed appropriately as participants changed their position in the environment and switched groups. In addition, an icon was placed above a participant's avatar, and by the side of their face on the HUD, when they were talking.

5.2.1.3 User Interface

The participants used desktop workstations, and a two-handed control method, with one hand on the keyboard and the other hand on a 3-button mouse. The user interface for the movement and basic MGD functionality (the MGDs from Chapter 4) was the same as before, outlined in Section 4.4.1.4. Figure 5.5 shows a reminder of the keyboard controls, with the addition of the 'numpad zero' key for the new functionality. The mouse move-

Figure 5.5: The keyboard controls for the teleporting and multiple views conditions. The new control is the 'numpad zero' key.

ment controlled heading and pitch, and the mouse buttons were used for basic MGDs techniques as per Section 4.4.1.4.

Holding down the numpad zero key released the mouse from controlling heading and pitch, and allowed it to control the position of the red teleporting arrow. Once the arrow was positioned in the desired location, a left mouse click teleported the participant there. This was designed to make it easier for participants to travel around the environment quickly (walking is time consuming).

The multiple views condition was designed to help participants understand their group members' perspectives. In addition, participants in the multiple views condition could position the teleporting arrow over one of their group member's views, and clicking the left mouse button would teleport them to that group member's location. This was particularly useful if that group member had seen something of interest—participants could take a closer look by teleporting to that group member, and seeing the point of interest in the main viewport. By default, the participant's subsequent movements were tethered to that group member (the automatic following functionality in basic MGDs) but the participant could 'free' themselves simply by pressing a movement key.

The multiple views took up the bottom quarter of the screen. A limit was imposed of three views, each taking up a quarter of the horizontal space, with the remaining quarter reserved for displaying the faces of any other group members (Figure 5.6). These could be selected using the numpad zero key to release the mouse pointer. Selecting them showed their view in one of the existing viewports, swapping out the member whose view had been replaced. The member who was replaced changed each time a swap was made—the system cycled through the group members in turn.

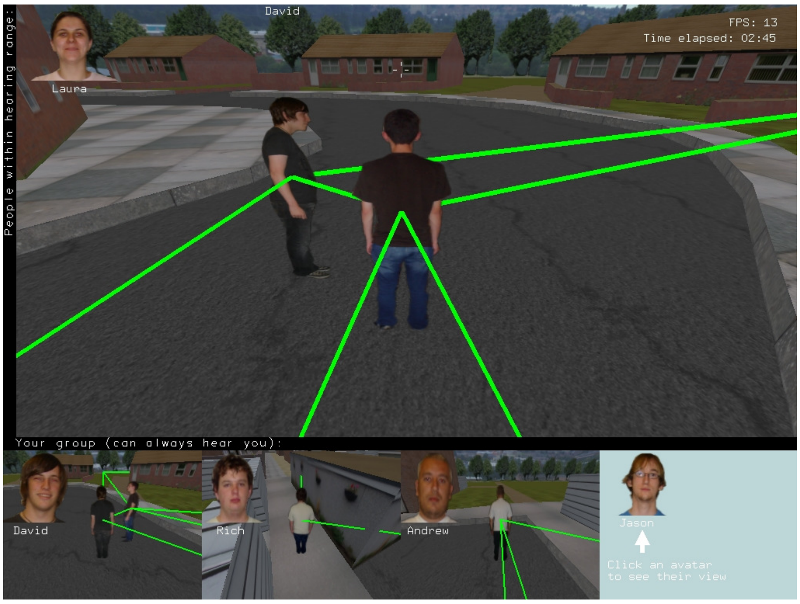


Figure 5.6: The multiple views condition, shown with a group of 5 participants. Three thumbnail views are shown, and the face of the remaining participant is displayed in the bottom-right. Clicking on this photograph showed their view in one of the existing viewports, and swapped out the member whose view had been replaced.

5.2.1.4 Procedure

The experiment used an urban planning scenario, outlined in Section 2.6. A 10 minute meeting was held with participants a few days before the experiment. They received a verbal explanation of the experiment, a single-sided A4 sheet containing extracts from UK urban planning guidelines (Appendix A), and a consent form. They also had photos taken for their avatar during this time.

The experiment itself lasted one hour. At the start participants were provided with another copy of the urban planning guidelines sheet, an instruction sheet for using the CVE (Appendix B), an experiment schedule, and an electronic copy of an urban planning report which they had to complete during the experiment. The questions for the report and the training process were the same as the previous study (Section 4.4.1.5), and the questions are listed in Section 2.6.

5.3 Results

There were two types of work that took place in the experiment: taskwork and teamwork [7], Section 2.5. Taskwork refers to the answers given in participants' reports, whereas data about teamwork were provided by the server's log of the movements, communication and groups that participants formed.

The urban planning reports were marked like an exam. Participants names were on the reports, marking wasn't blind. An independent samples t-test showed no significant difference between the teleporting and multiple views conditions, $t(13) = 1.49, p = .16$. Participants in the teleport condition had a mean mark of 18.7 ($SD = 3.3$) out of 24, and 16.3 ($SD = 3.1$) in the multiple views condition. The focus, however, was on how participants went about doing the task (i.e. the teamwork), and how different MGD functionality affected participants' behaviour. This was analysed both quantitatively and qualitatively.

5.3.1 Quantitative Analysis

For each batch of participants, the spoken and text communication was transcripted and analysed using a communication coding approach [15] to classify each utterance as in Section 4.5.2.1.

These data were analysed in terms of the quantity of communication that took place, and where participants were relative to each other when they communicated. For comparison, data are provided from the previous study (Chapter 4) when other participants had performed the same urban planning task either in a conventional CVE ('control' in Figure 5.7) or with basic MGDs functionality (see Figures 5.7 and 5.8). Note that the average group size in the basic MGDs, teleport and multiple views conditions was 3.5, 2.5 and 3.0 respectively.

The total number of utterances made by participants in the basic MGDs (data from Chapter 4), teleport and multiple views conditions (data from the present study) was analysed using a univariate analysis of variance (ANOVA). The Kolmogorov-Smirnov test showed that the data in the three conditions were from normally distributed populations, $D(8) = .171, p > .05$ (basic MGDs condition), $D(7) = .255, p > .05$ (teleport condition), $D(8) = .209, p > .05$ (multiple views). Levene's statistic showed that variances between the conditions were significantly different, $F(2,20) = 7.71, p = .003$, and so a Brown-Forsythe F-ratio was used for the ANOVA (the Brown-Forsythe F-ratio was derived to be robust when the data do not have homogeneity of variance).

This showed that there was a significant difference between the conditions, $F_{BF}(2,13.6) = 3.92, p = .045$. The mean amount of communication increased by 226% from the basic

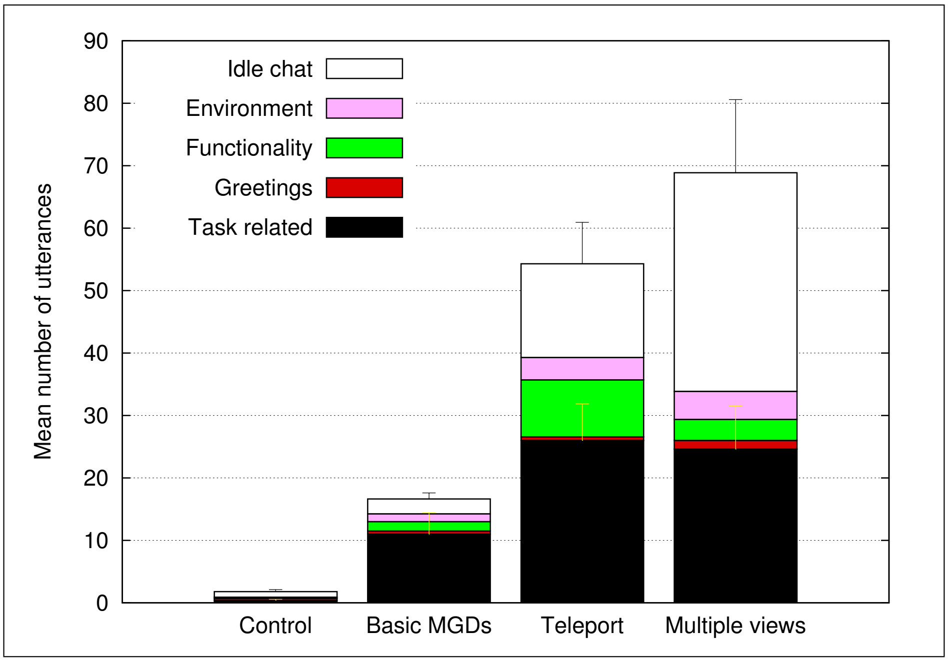


Figure 5.7: Mean number of utterances made by the participants in each condition. The control and basic MGDs conditions are from Chapter 4. The error bars are shown for task related utterances and idle chat.

MGDs to the teleport condition, and by another 27% from the teleport to the multiple views condition. Within this, task related communication increased by a factor of two from basic MGDs to the teleport and multiple views conditions, but this was not significant. Idle chat more than doubled from the teleport to the multiple views condition (see Figure 5.7).

One of the limitations identified in the previous study (Chapter 4) was that participants tended to assemble in one place in the CVE before communicating, even though this was unnecessary with the basic MGDs functionality that was provided (Section 5.1). To determine whether the new functionality provided in the present study overcame this limitation, each time a participant made an utterance the distance to their nearest group member was calculated, and the mean for each participant in the basic MGDs, teleport and multiple views conditions was analysed using a univariate ANOVA. The two participants who didn't speak at all during the experiment were excluded from the analysis, one was from the basic MGDs condition and the other was from the multiple views condition. The ANOVA showed that there was a significant difference between the conditions, $F(2,18) = 3.56, p = .05$. Tukey HSD posthoc tests showed that the difference between basic MGDs and multiple views was significant ($p = .04$) but the other pairwise comparisons were not (see Figure 5.8).

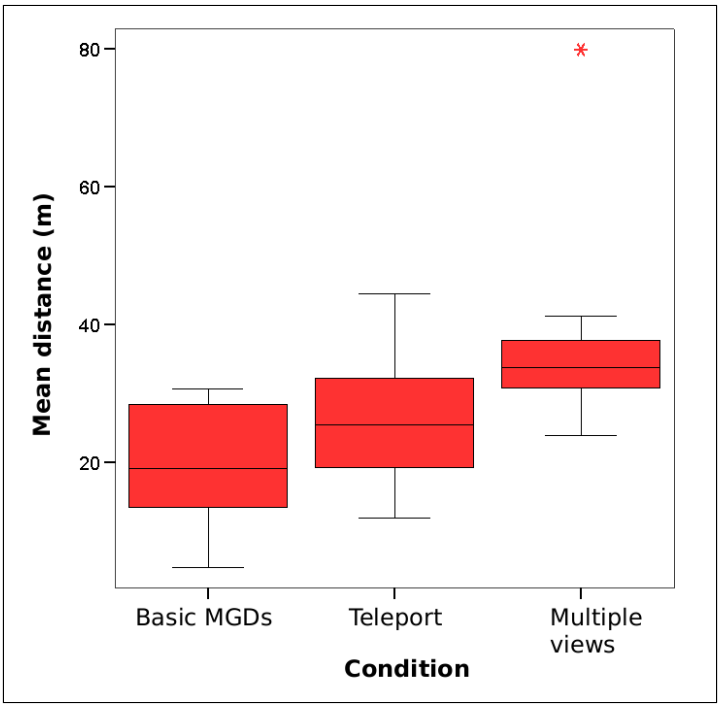


Figure 5.8: Mean distance to the nearest group member at the time of each participant's utterances. The basic MGDs condition was from Chapter 4.

5.3.2 Qualitative Analysis

The quantitative analyses show that teleporting and multiple views increased both the quantity of communication that took place and the distance over which participants communicated. The purpose of the qualitative analysis was to understand the underlying behavioural changes that cause these increases, and how teleporting was used in general.

The server log allowed the distances participants travelled while teleporting and walking to be calculated and showed that, overall, 16% of travel was by teleporting. Further investigation showed that there were two distinct uses of teleporting. First, teleporting was used to speed up exploration of the environment, particularly when participants first entered the environment (see Figure 5.9). Second, teleporting was used to reach points of interest. For example, at one point during the experiment the some participants' conversation was about blank walls, which was relevant to one of the questions in the task. The blank walls were at the ends of the horseshoe-shaped buildings. Participant *O*, represented by a green line in Figure 5.10, teleported across the building on the left to view the blank walls (timestamp [26:20]). *O* then teleported up to the top of the map to see the walls that *I* and *R* were talking about.

In order to teleport to a point of interest, a participant must first know its location within the environment. This is sometimes difficult, as the following conversation extract

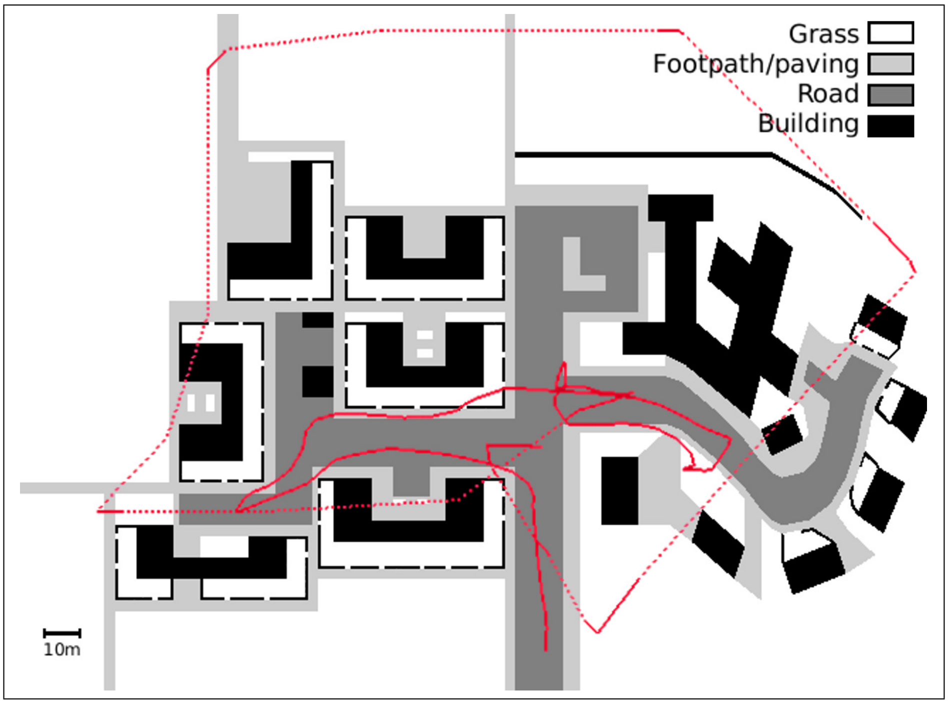


Figure 5.9: Path showing the first 5 minutes of movement of a participant who used teleporting to speed up their exploration of the environment. A solid line represents walking, and a dotted line represents teleportation.

from the teleport condition shows:

> [09:25] O: Look, show me, *M*, show me the two entry points then, the road ones at least.
>
> [09:29] M: Alright well, are you where I am now?
>
> [09:35] O: Where are you?

The multiple views condition helps with this by allowing participants to teleport to the location of a group member by clicking on their viewport:

> [39:42] C: That's useful!
>
> [39:43] S: Yeah! Where are you? I'll show you!
>
> [39:49] C: I'll teleport to you! Hang on!

Each component of the new MGDs functionality (awareness of who could hear one's communication, multiple views and teleporting) that was provided in the present study had the potential to increase the distance over which participants communicated. The data indicate that multiple views made the greatest contribution (see Figure 5.8). To identify whether awareness or teleporting was the most important secondary cause a detailed analysis was made of the communication and movement of the two participants (*I* and *O*) who spoke the most in the most persistent group in the teleport condition.

I and *O* both spoke in 18 conversation blocks, but used teleporting in only four of these blocks. On all four of these occasions, *I* and *O* used teleporting to collocate within the environment. In the other blocks *I* and *O* either were together (5 blocks), remained separated (2 blocks), separated without teleporting (3 blocks) or collocated without teleporting (4 blocks). This suggests that the awareness functionality was more important than teleporting for increasing the distance over which participants communicated.

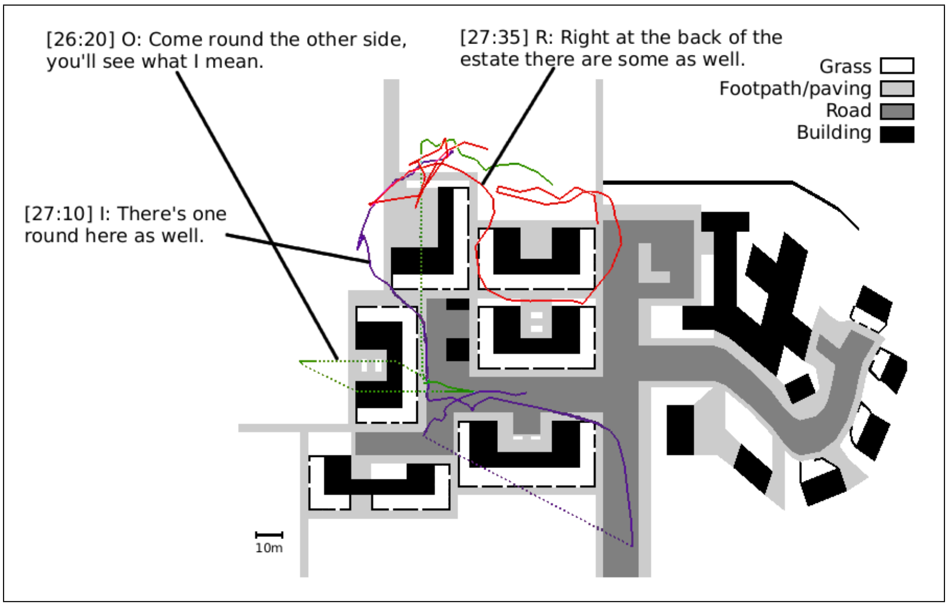


Figure 5.10: Paths taken by participants *O* (green line), *I* (purple line) and *R* (red line) when talking about blank walls (a point of interest). The solid lines representing walking, and dotted lines represent teleporting (participant *O* teleports to points of interest at the ends of the horseshoe-shaped buildings). To place participants' movement in context, the paths are labelled with timestamps and conversation utterances.

5.4 Discussion

It is well known that in conventional CVEs users often have difficulty understanding the context of what each other is talking about (see problems 1 & 2 in Section 2.3.2.1). The previous research into MGDs (Chapter 4) showed how a group graph metaphor could help users find each other, since the graph 'tracked' participants and the nodes corresponded to avatars, with edges denoting group membership (see Figure 5.1). One could find a group member by following a line from their avatar until they reached a node. In the present study, the teleporting and multiple views functionality took this a step further. It allowed participants to teleport directly to a group member of their choice by selecting the appropriate viewport, and the qualitative data gave an example of how this helped participants (see the conversation extract in Section 5.3.2). Furthermore, providing participants with multiple views specifically tackled the problem of understanding another's perspective.

The qualitative data showed that teleporting was used in two ways: increasing the speed of movement (in particular, an initial speed search) and movement to points of interest. This is a simple method of time-saving functionality, however potential drawbacks must not be overlooked. Firstly, as with any new functionality, a potential problem might be making the system over complicated. Other features may be forgotten about and may not be used to their full potential. Secondly, and perhaps more subtly, teleporting could mean people lose the feel for distance. In an urban planning context, it is important that participants in the CVE get a feel for the scale of the environment, in particular the size of buildings and proposed developments. One of the questions for the urban planning report was asking participants if they thought the blocks of houses were the right size. Financial savings are made by building the houses joined together in blocks, but a large block size decreases permeability of the estate, making it bad for transport and pedestrians (they have to go further before they can change direction). Teleportation may mean participants lose a sense of scale and large blocks could go unnoticed.

One of the places where the original MGDs techniques fell short of their goals was in facilitating communication when participants were spatially separated within the environment. The fact that participants tended to collocate to communicate in the basic MGDs condition was a sign of inefficient groupwork—participants were either taking time to collocate when they wanted to communicate, or they were waiting until they were coincidentally collocated before they said anything. Providing functionality to communicate with group members from a distance, and informing the participants of this in the instructions, was not enough to make their behaviour more efficient. This study indicates that by providing feedback to the participants, they became more aware of how the system works. The quantitative data showed that in both the teleporting and multiple views conditions, participants communicated across greater distances than in the basic MGDs condition (see Section 5.3.1). The interesting thing about this feedback from the system is it's not specifically new functionality in the sense of a new tool at the users' disposal, like teleporting and multiple views are. Instead it provides awareness of *existing* functionality: the ability to communicate with group members from a distance. As Schroeder et al. reflect, does one improve usability 'by means of improving the systems and features of the environment, or by improving the users' awareness of their activities and settings?' [96, p. 666].

Finally, in previous research, participants communicated a great deal to overcome the lack of sensory information that CVEs provided [51] [90]. However, in the present study participants communicated much more than in conventional CVEs because they were provided with more sensory information (e.g. awareness of who could hear you and who was speaking, and multiple views providing an 'extra pair of eyes'). The quantitative data showed that the amount of conversation increased 4 times from the basic MGDs condition to the multiple views condition. This increase in communication was indicative of more teamwork taking place.

Chapter 6

Virtual Time

6.1 Introduction

Collaborative work can be synchronous (where participants are working at the same time) or asynchronous (e.g. shift work, leaving voicemail messages, communicating by email, see Section 1.1).

Chapter 4 describes the implementation and evaluation of techniques that support synchronous group work in CVEs. These techniques are known as Mobile Group Dynamics (MGDs). Chapter 5 addresses the shortcomings in the original MGDs and discussed the development of new functionality to improve teamwork when people were collaborating over extended distances—that is, they were spatially separated in the virtual world.

This chapter describes the implementation and evaluation of a new concept called Virtual Time (VT), that facilitates asynchronous collaboration. VT allows (to a certain extent) virtual synchronisation of people who are physically separated in time. Taken together, the MGDs and VT techniques allow both synchronous and asynchronous collaborations in large-scale CVEs.

6.2 Methods for Virtual-time 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. Section 2.4 discusses related work regarding asynchronous collaboration in virtual environments.

If one considers a spoken or written utterance to be the basic unit of collaboration, a basic VT system would just contain what was said in that CVE, but nothing about who said them, what they were talking about, where they were in the CVE, or when. At the other extreme, a sophisticated VT system would allow you to travel through a virtual world, walking with people who had been there in the past, chipping in to their conversations as if they were still there, to the extent that an observer who came along later still would be unable to determine who were the original inhabitants versus who was the impostor who'd been added later?

Analysis of these examples highlights a rich complexity of possible functionality for VT. Therefore, the following sections present a framework of VT, and then describe the practicalities of implementation.

6.2.1 A framework for virtual time

Given that utterances are the basic building blocks of collaboration and communication in virtual worlds, then a key challenge for virtual time is determining how those utterances should be organised and associated. The primary methods for doing this are in terms of: (a) people, (b) time, (c) space and (d) topic. Each method has several levels (see Table 6.1), which can provide context to help us understand the meaning of what was said, influence where we choose to go next in the CVE, and help users control the number of utterances that are visible/audible at any given time so the VT system is scalable.

Adding people's identity to the utterances in a VT CVE allows users to discriminate everything that was said by a particular person, for example, someone who provided particularly insightful comments. Allowing people to choose their virtual appearance will have other effects on whose utterances a given user chooses to listen to, as outlined in Section 2.3.2.1.

Statistics terminology is adopted for the levels of time. At the nominal level, a future user would have no clue as to when, or in what order, different utterances were spoken. Ordinal information would allow utterances to be listened to in the sequence that they originally occurred, and the time interval (either absolute or rebased to when the speaker entered the environment) would allow sets of utterances that took place in quick succession to be distinguished from those that were separated by a lengthy delay.

Indicating the point in space where each utterance was spoken would help a future listener understand what was being talked about, and reduce the need for users in CVEs to devote much more effort to making the ‘implicit explicit’ than is the norm in real life [51], [90]. Linking utterances by the path the speaker had taken would provide the listener with even more information about the things the speaker had seen and which led them to a particular conclusion.

Organising and associating utterances can be done according to certain topics (or subtopics), for example defining whether greetings were due to users meeting or departing, whether idle chat was humorous or not, or which part of a task a given conversation was based on. Natural Language Processing (NLP) algorithms could be used to process utterances into topics, to which user-supplied quality ratings could be added by borrowing techniques from recommender systems and search engines.

Method	**Level 1**	**Level 2**	**Level 3**
a) People	Anonymous	Identity	Appearance
b) Time	Nominal	Ordinal	Interval
c) Space	Amorphous	Point	Path
d) Content	Undefined	Topic	Quality

Table 6.1: Four methods for organising/associating utterances. Level 1 corresponds to a basic VT system, with Levels 2 and 3 providing ever richer possibilities for virtual time.

Finally, there are many possible combinations of the above methods. For example, combining point (space) and topic (content) would allow the main items of interest in a given area to be quickly determined, adding interval (time) to identity (people) would allow the conversations of a group of people to be followed, and adding path (space) to interval/identity would help a future listener comprehend the bigger picture of a conversation that took place after a group of people had split up to explore an area and then regrouped to discuss their findings.

6.2.2 Implementing virtual time

For the evaluation described in this chapter, virtual time was implemented using level 3 utterance association for people (appearance) and level 2 association for time (ordinal), space (point) and content (topic) (see Table 6.1). Details of the implementation are as follows.

First, all of the utterances from the two previous studies of synchronous teamwork in CVEs (Chapters 4 and 5) were divided into blocks of communication and categorised as either task-specific or not. The latter were discarded to avoid cluttering the CVE with irrelevant utterances (e.g., idle chat). The task-specific blocks were classified using keywords from the 13 questions on the urban planning report that participants were asked to complete (see Section 2.6) and, although this was performed manually in the present study, it could have been done using NLP techniques.

The classification used a two level hierarchy, with the 13 questions (subtopics) grouped according to three topics (*permeability*, *character*, and *safety and security*) that were used on the urban planning report. The topics were rendered with different hues (yellow, cyan and magenta), using a different lightness for each subtopic (Figure 6.1). A colour-coded tick box was provided for users to choose which utterance subtopic(s) were displayed (Figure 6.2), allowing related comments to be identified even if they are separated in space and time. In addition, tags flashed when they were being played, and were visually caged in black stripes when they had been viewed (Figure 6.3).

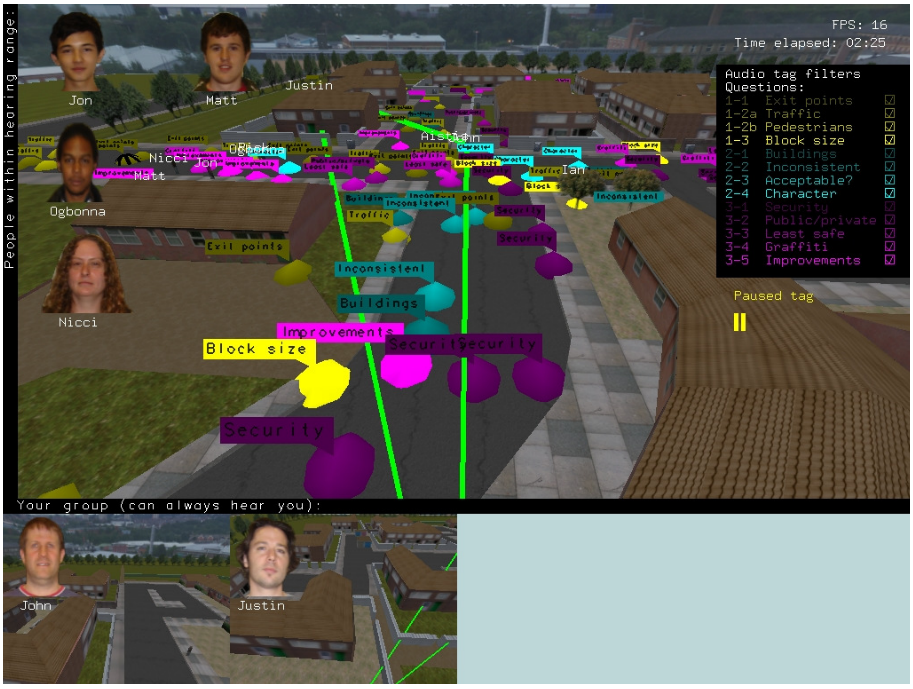
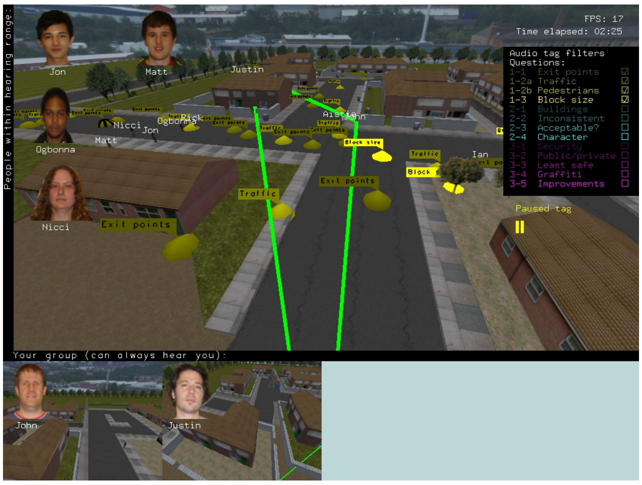
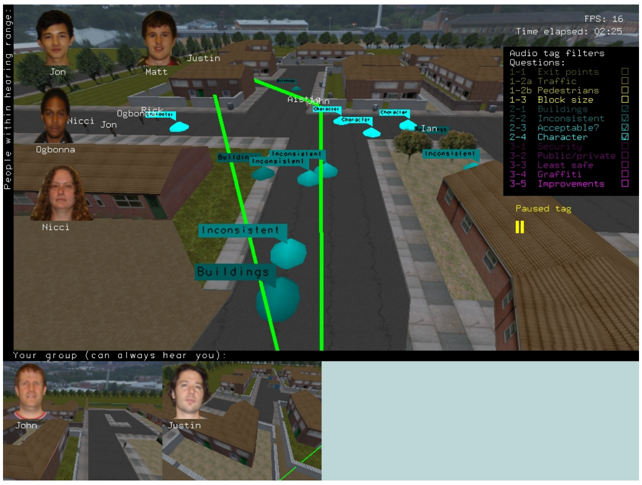


Figure 6.1: All of the utterance tags used in the VT study. The 'Audio tag filters' was a list of tick boxes, shown in the top right hand corner of the screen, that allowed the utterances associated with each of the 13 questions to be toggled on/off.

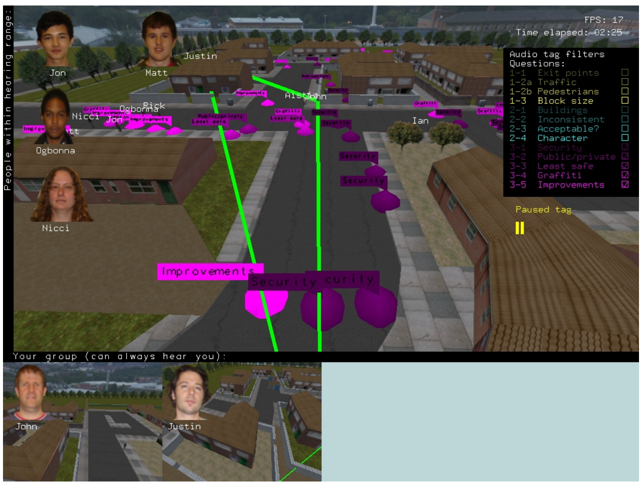
The point where each utterance commenced was represented with a hemispherical visual object known as a tag, which put the conversations into context by showing where they took place (Figure 6.4). To reduce clutter, all utterances in a given block that were within line of sight of each other were represented by a single tag that was at the mean position of the individual utterance tags (Figure 6.5). This reduced the overall number of visual tags from 160 to 96. These tags represented 67 conversation blocks (some conversations had more than one tag associated with them, e.g. speakers at different parts



(a) The first main topic ('Permeability').



(b) The second main topic ('Character').



(c) The third main topic ('Safety and security').

Figure 6.2: The utterance tags shown with filters applied to show the three main topics.

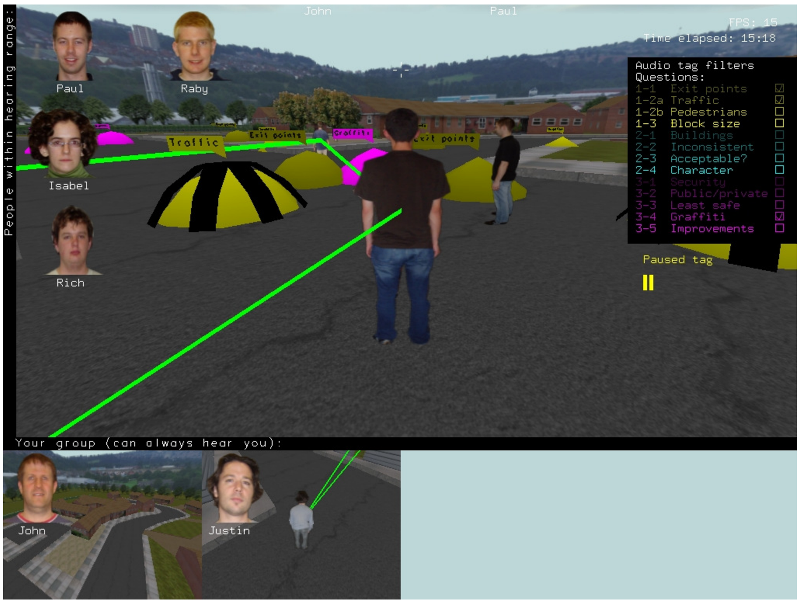
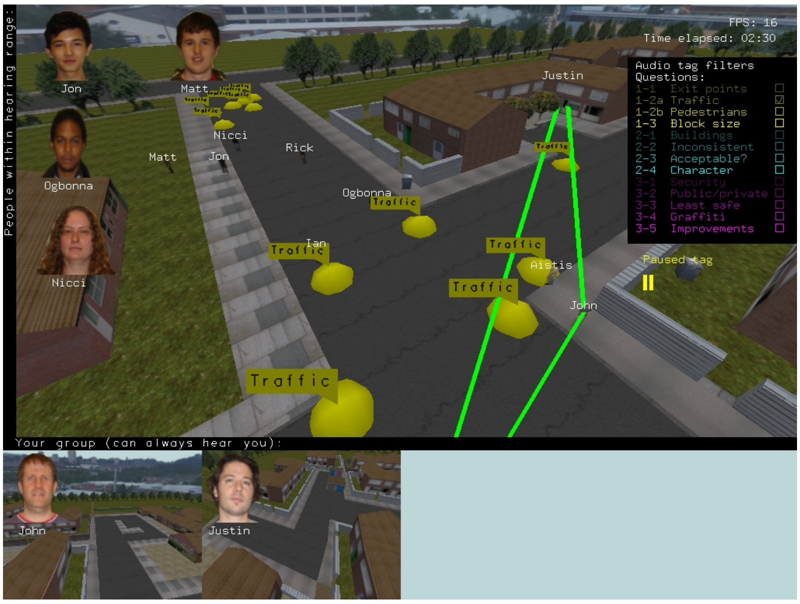
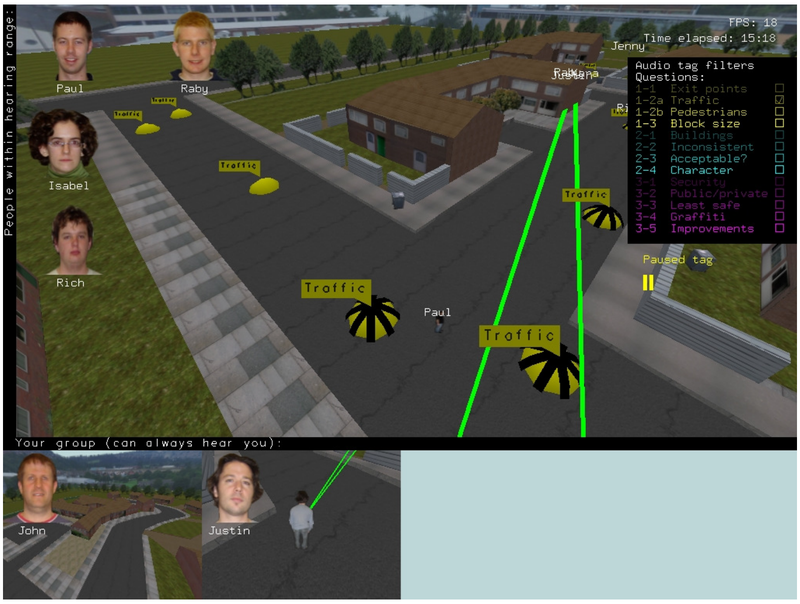


Figure 6.3: The utterance tags shown from an over-the-shoulder perspective, with three subtopics selected. Utterances that a pair of participants had already listened to were visually caged in black stripes.

Figure 6.4: The utterance tags were positioned at the point where each utterance commenced, which was designed to allow participants to put the conversations into context. This screenshot gives an example of a conversation on the subtopic of 'public/private spaces', which was put into context by its position in the front gardens of some terraced houses.



(a) The subtopic 'Traffic' without line of sight reduction applied. This is shown for comparison with (b) (the utterance tags were not presented in this way to participants).



(b) The subtopic 'Traffic' with line of sight reduction (as the participants in the study saw them). The tags that had been listened to by participants were visually caged in black stripes.

Figure 6.5: The utterance tags, shown with the filter applied to only show the subtopic 'Traffic'. The tags are shown here with and without line of sight reduction, for a comparison. In the environment used for the study, the tags that referred to the same conversation and were within line of sight of each other, were replaced by a single tag at the mean position, to reduce clutter. Utterances that a pair of participants had already listened to were visually caged in black stripes (b).

of the environment without line of sight would have each been given a separate tag object at their location).

The system was tested using a pilot study and refined in response to participants' feedback. The main improvements were:

- Making tag selection explicit (instead of walking into a tag to play it, the users wanted to be able to select a tag with the mouse).
- Allowing users to pause/resume/stop tags, instead of always playing the whole of a tag.
- Providing more time for the task than was allowed in the previous studies (Chapters 4 and 5), because there were a lot of recorded utterances that the users wanted to watch and listen to.

6.3 Hypothesis

Due to the exploratory nature of this work (few observations of asynchronous teamwork in virtual environments), it was difficult to generate meaningful hypotheses for the way participants would use the VT system and the changes it would make to their behaviour. Conversation tags were designed to improve teamwork by making available the activities and conversations of others to help them. The metrics used were the usage of the conversation tags, the performance on the task and the distance travelled. Whilst differences between the conditions were expected, it was difficult to predict the direction of change: would participants communicate more now they had more information available to them, or less now that they could just listen to the tags? Would participants travel further as they were visiting all the tags, or not as far, since they did not have to spend as much time navigating the environment for themselves when the answers were on-hand? The experiment was a first step in analysing asynchronous collaboration in CVEs, and the results will help researchers make more informed design decisions and predictions in future work.

6.4 Virtual-time collaboration

The experiment used the same urban planning context and the same environment as the previous studies (Section 2.6). Participants were run in pairs, and had access to all the task-related conversations of 23 people who had previously done the same task in the environment (the 8 participants in the basic MGDs condition from Chapter 4, and the

15 participants of the teleporting/awareness and multiple views conditions in Chapter 5). These previous conversations were embedded in the environment using the tags described in Section 6.2.2.

6.4.1 Method

The method was similar to that of the previous studies (Chapters 4 and 5, see Section 4.4.1). A total of 10 participants (5 pairs) took part. There were 7 men and 3 women, with a mean age of 22.2 ($SD = 3.3$). They had not taken part in any of the previous studies. Each pair communicated with each other and had access to the task-related conversations of 23 participants who had done the task in the previous real-time MGDs experiments.

After pilot testing, the total time for the experiment was increased to 90 minutes from the 60 minutes used in the previous studies. The time in the training environment was extended from 15 minutes to 30 minutes, and the time in the residential environment from 35 minutes to 45 minutes. The final 15 minutes were allocated for a semi-structured interview.

The grouping interface and functionality were identical to that of the multiple views condition (Chapter 4, see Figure 5.4), however each pair was placed into a group together and could not join the groups of participants from the past. Participants could select a conversation tag by positioning the crosshair with the mouse and pressing the left mouse button. Participants could stop conversation tags by pressing the 'Escape' key, and pause/resume the playback with the 'F1' key.

6.4.2 Results

As in the previous studies (Chapters 4 and 5), the urban planning reports were marked like an exam, and participants had a mean mark of 17.9 out of 24 ($SD = 3.9$). An independent samples t-test showed no significant difference between the teleporting and multiple views conditions, $t(16) = 0.978, p = 0.343$. However, the main interest lay in how participants used VT and the effect it had on their behaviour in the CVE. To investigate this, participants' communication, movement and tag usage were analysed. Statistical comparisons were made with the multiple views condition from Chapter 5, whose interface was the same except for the VT functionality. In considering the findings, readers should bear in mind obvious differences between the experiments (especially the number of live participants at any given time), which could have affected the results.

Each participant's rate of communication was calculated by dividing the number of utterances they made by the time they spent working in the CVE (the difference between the time they first and last moved). This took account of the extra time allowed for the VT study as a whole (45 vs. 35 minutes) and the fact that some participants remained 'in' the CVE (but not moving) while they finished writing their report.

Figure 6.6 shows the mean rate of communication for task-related and non-task-related utterances for the multiple views (Chapter 5) and virtual time (VT) groups. An independent samples t-test was carried out on the task-related communication, and showed that the difference between the groups was significant, $t(16) = 3.258, p = 0.005$.

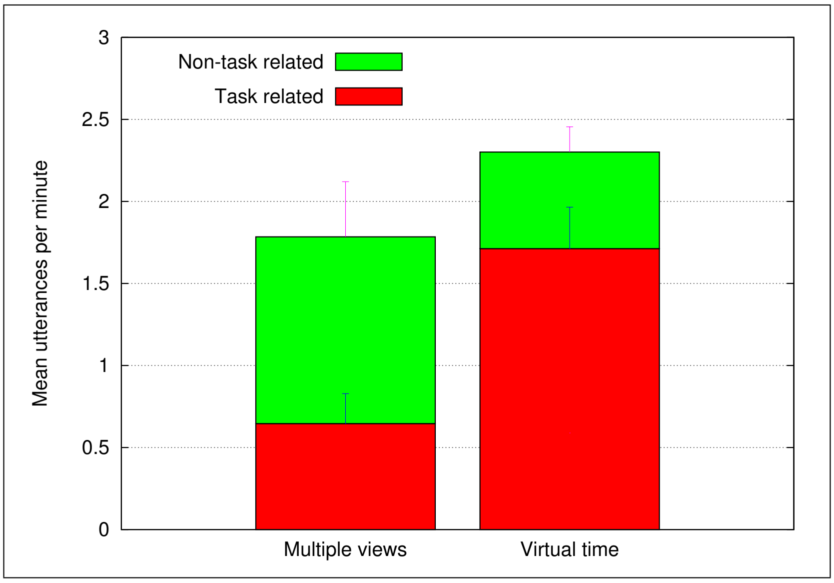


Figure 6.6: The mean utterances per minute for the multiple views and virtual time conditions.

The distances that participants covered as they walked and teleported around the environment were calculated from the server log. The rate of travel was calculated by dividing the distance each participant travelled by the time they spent working in the CVE (calculated as above). Independent samples t-tests showed that the multiple views group walked significantly further in unit time than the VT group, $t(16) = 2.790, p = 0.013$. The difference in rate of travel for teleporting was not significant, $t(16) = 0.578, p = 0.571$.

The paths of two participants were plotted as a qualitative analysis of the usage of teleporting. The two participants chosen had the median percentage distance teleported. The paths are shown in Figures 6.8(a) and 6.8(b). The paths show examples of how teleporting was used: to cover large distances and to 'jump' over buildings.

There were a total of 67 conversations tagged. All of participants' usage of the tags, and the tag filter menu, was recorded in the server log. Analysis of this log showed that pairs of participants typically selected one subtopic in the menu at a time, allowing participants to focus on VT utterances that were relevant to the question being answered at

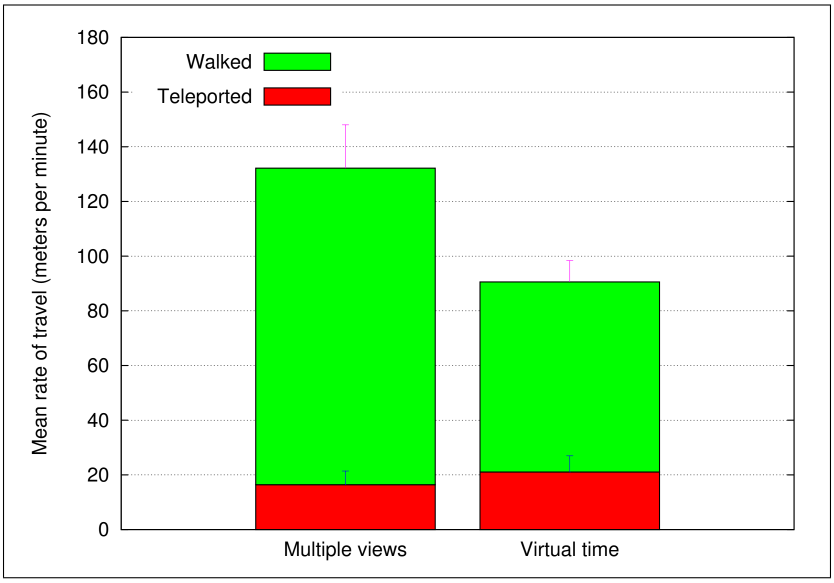


Figure 6.7: The mean rate of travel for the multiple views and virtual time conditions.

a given time, and went through the tags in a logical order (subtopic by subtopic, matching the order in which the questions appeared in the urban planning report).

The mean number of tags played by each pair was 20.0 (SD = 6.6), and the breakdown by (sub)topic is shown in Table 6.2 (note: the playback was shared across the network, so both participants in a pair heard the same utterances). The distribution of the utterance tags and the frequency with which each was played is represented in Figure 6.9.

6.5 Discussion

In this study, VT was implemented via a system of conversation tags (Sections 6.2.1 and 6.2.2) so participants could take advantage of the comments their predecessors had made. Participants used the environment in pairs, so each had one other real-time collaborator to communicate with and the conversations of 23 previous inhabitants to listen to. The results showed that the VT system led to a significant increase in task related communication between the 'live' pair when compared to the same interface without the VT (the multiple views condition). In other words, virtual time stimulated communication between the live participants. Further, the results showed significantly less travel around the environment.

The reduced travel suggests that the points of interest were found by watching and listening to the conversations from the past: the usage results showed that on average a quarter of the conversation tags available for each subtopic were played. There is one

Figure 6.8: The paths of two participants who had the median percentage distance teleported. The solid lines represent walking and the dotted lines represent teleporting.

Figure 6.9: The environment with the utterance tags. The tags are coloured based on how many pairs of participants played them: white = never played, blue = played by the minority of pairs, and red = played by the majority (3+).

Topic	**Subtopic**	**Total tags**	**Mean tags played**	**SD**	**%**
Permeability	Exit points	18	4.80	3.49	26.7 %
	Traffic	10	4.20	2.39	42.0 %
	Pedestrians	0	0.00	0.00	0.0 %
	Block size	4	2.00	1.58	50.0 %
Character	Buildings	3	1.40	0.89	46.7 %
	Inconsistent	5	0.80	0.84	16.0 %
	Acceptable?	0	0	0	0.0 %
	Character	4	2.00	1.41	50.0 %
Safety & security	Security	10	1.40	1.52	14.0 %
	Public/private	3	1.00	1.22	33.3 %
	Least safe	2	0.40	0.55	20.0 %
	Graffiti	4	0.80	1.30	20.0 %
	Improvements	4	1.20	1.30	30.0 %

Table 6.2: The mean number and SD of utterance tags in each subtopic that were played by each pair of participants.

drawback to this, however, as identified from the participant's comments in the semi-structured interviews. They were concerned about the quality of content of the tags, since they were using the information from past participants to perform the task. Assuming the past work was thorough and correct, VT provides a large pool of ideas to be shared from one group of participants to the next, thus focusing their conversation on the task. This highlights the importance of a measure of quality of content, level 3 in the VT framework, Section 6.2.1. For example, tags could be user-generated, so that participants could explicitly request that particular conversations be recorded and tagged, a user-rating system would allow participants playing back the conversations to contribute to the quality control, and methods from recommender systems (e.g., Amazon's 'people who bought X also bought Y') could be incorporated.

Chapter 7

Conclusions

Collaborative interaction can be classified in terms of time and space (Section 1.1, [56]). CVEs typically allow synchronous interaction (time), between people who are physically separated (space). However, this thesis describes the development of techniques to support both synchronous and asynchronous interaction in CVEs.

The overall goal was to make collaborative interaction in virtual environments more effective, more like that of face-to-face interaction, without necessarily imitating reality, or unnecessarily limiting ourselves to naturalistic constraints (cf. [79], [39]). Potentially, a virtual environment is so flexible it could be a more effective place for teamwork to be carried out than the 'real world' (a hypothesis shared by Linebarger et al. in [66]). In addition, CVEs can enable collaboration from a distance (over the Internet), and if the technology was good enough it would reduce the need to travel large distances in the real world (saving businesses time and money, helping distributed communities stay in touch, and reducing our impact on the world's environment). There are limits, of course, and some problems caused by distance cannot be eradicated with technology (e.g. time zone, cultural differences [77]).

The research began with a suite of techniques called Mobile Group Dynamics (MGDs). These techniques supported the first and last stages of Tuckman's *forming*, *storming*, *norming* and *performing* model [116]. MGDs enabled people to form explicit groups (using spatial positioning and selection), and these groups were represented using a novel graph metaphor (the avatars of each group were connected by lines that tracked them as they moved around the environment). Each graph was given a unique colour to identify that group (Section 4.2.1, Figure 4.2).

MGDs provided functionality to aid the performing of activity in three ways. First, by allowing people to select a group member to automatically follow, so that participants could travel around the environment together. Second, a group member could remain in communication with the rest of their group over any distance (other audio communication was distance attenuated to simulate real world verbal communication). Third, the group graph was designed to provide an awareness of fellow group members' activities, since they could be located by following a line that connected your avatar to theirs.

The functionality was evaluated by comparing an urban planning task performed with the MGDs techniques to a CVE with conventional functionality (avatars, audio/text communication, but no explicit groups, and no automatic following mechanisms). The MGDs techniques produced fundamental changes in participants' behaviour: with MGDs they communicated seven times more and spent twice as much time in close proximity (Figures 4.9 and 4.10).

A qualitative analysis looked at patterns of interaction by drawing the paths of participants and listening to their conversations at that time. (The software application was programmed to automatically draw the paths of participants onto a map of the environment, and the conversations were timestamped and transcribed by hand). The resulting images showed evidence of real world behaviour, and participants were observed to be moving together (using the automatic MGDs functionality), splitting up to explore the environment separately, regrouping, and navigating together to a point of interest (Section 4.5.2.2, Figure 4.11).

The group graph and the automatic following functionality specifically helped with the observed imitation of real world group dynamics, since these techniques made it easier for groups to be identified (the group graph), remain together while they travelled (the automatic movement), and regroup after periods of separation (an alternative way of using the automatic following functionality). This behaviour contrasted with the control condition, where participants had larger spatial separation and relatively low communication, which is evidence that they primarily worked as individuals when they weren't provided with MGDs (a level 2 analysis, Section 2.5.2).

MGDs allowed communication over an infinite distance between group members, but despite this participants tended to spatially regroup to discuss their findings. In addition, if they wanted to talk about a particular point of interest in the environment, they had to 'walk' to the appropriate location. Regrouping and navigating to a point of interest take unnecessary amounts of time. Chapter 5 describes three improvements on the original MGDs techniques that were developed to tackle these issues.

First, participants were allowed to teleport to any part of the environment by clicking on it.

Second, participants were provided with 'awareness' of existing functionality, by placing the faces of participants who were within hearing range onto their view of the environment, using a HUD metaphor. This allowed participants to see who would hear them when they spoke—the faces of people nearby would appear and those out of range would be removed as participants travelled around the environment, but the faces of fellow group members would always remain. A speech icon was placed next to the face of a person who was talking to provide further visual feedback, useful for when their avatar was out of sight.

Third, participants were provided with multiple views of the environment. Participants had their own main view, and this was augmented with thumbnail views of their fellow group members. This enabled participants to see what others were looking at in real-time, and in addition they could click on a thumbnail view to teleport to be next to that person.

These additional techniques were evaluated by dividing new participants (they had not used the basic MGDs from the previous study) into two batches. The first batch of participants had all the basic MGDs techniques from the previous study, with the addition of teleporting and awareness functionality (the 'teleporting' condition). The second batch had all the functionality of the first batch, with the addition of multiple views (the 'multiple views' condition).

The paths of participants were plotted to provide a qualitative analysis, and the travel by teleporting was distinguished from walking. The resulting images showed that teleporting was used for initial speed searches of the environment, and moving to points of interest (Section 5.3.2, Figures 5.9 and 5.10).

Teleporting is an obvious time-saver to have in a CVE system. A disadvantage might be that users lose a sense of scale (which might be a problem in the context of reviewing an urban planning design, see Section 5.4). Further research would be required to determine if that is a genuine problem.

A quantitative analysis showed that the new functionality helped facilitate communication when participants were spatially separated within the environment. The mean distance between participants and their nearest group member at the time of each of their utterances increased with the new functionality, and was statistically significant between the basic MGDs and multiple views conditions. Further, the mean amount of communication increased significantly (see Figure 5.7).

These results showed that the new functionality helped participants take full advantage of the system (saving time by not collocating to communicate). The ability to communicate across large distances was already there, but the new functionality provided awareness of it (identifying who was within hearing range and who was talking). The multiple views provided awareness of the activities of fellow group members (were they moving, teleporting, in bird's-eye view, over-the-shoulder view, what were they looking at?) and solved the problems of understanding another's perspective (problems 1 and 2, Section 2.3.2.1).

Chapter 6 introduces a new concept called Virtual Time (VT), which was designed to facilitate asynchronous collaboration in CVEs. The movements and conversations of participants who had performed the urban planning task in the previous studies had been recorded to disk. The task-related conversations were given a hemispherical tag in the environment, and labelled with the topic of conversation (the urban planning question they were discussing). Figure 6.1 shows these tags, and people using the VT system could filter them by topic to reduce clutter (Section 6.2.2).

Participants could click on these tags using the mouse. Upon doing so, the avatars of previous participants that they referred to appeared in the environment, and their movements and conversations were played back as if they were there in real-time.

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 travelled around, but an increase in live communication that took place. Taken together with the MGDs, the studies show how CVE interfaces can be improved for synchronous and asynchronous collaborations, and highlight a number of possibilities for future research.

7.1 Future work

The evaluations in this thesis used an urban environment based on a real residential estate. However, as Harrison and Dourish argue, one cannot simply replicate a space from the real world in a CVE and expect real world behaviour to emerge [47]. The technology and functionality collectively form a part of the space, and contribute to the emergence of a sense of 'place' within the environment. Put simply, the functionality provided to participants has an affect on their behaviour.

CVE designers must not think of their work as designing a space for users to interact, but need to construct a combination of functionality that allows more versatile flows of information (e.g. navigation, communication), and feedback provided to the user to give them *awareness* of the functionality available to them (Chapter 5). The flexibility afforded by the CVE designer allows for many possibilities for future work, and these are discussed in the following sections in the context of MGDs (synchronous interaction) and VT (asynchronous collaborations).

7.1.1 Mobile group dynamics

Chapters 4 and 5 researched a variety of techniques to support synchronous interaction. There are several ways future work can build on this. First, the task performance in the studies in this thesis did not improve significantly, and it could be argued that this is due to the task being subjective. The techniques could be applied to new application areas, such as safety training, or online games. Both of these application areas can have objective goals (e.g. time taken to evacuate a building), and would provide an objective measure for performance. The hypothesis would be that MGDs would improve the task performance due to the improvements in teamwork evident from the studies in this thesis.

Would the MGDs results generalise to other application areas? Generalisability is a problem in psychological studies of interaction in a controlled setting, since the psychologists want to know if the results generalise to the real world, outside of the laboratory conditions. In the case of the type of experiments described in this thesis, Schroeder et al. point out that the results are not meant to transfer to the real world (from a controlled environment to one with lots of new parameters), but to transfer from one controlled environment to another (one CVE to another CVE with similar constraints) [96]. When viewed in this context the problem of generalisability is not as complex as it might originally sound—the hypothesis (above) remains intact.

Second, the experimental studies could be improved by learning about the effects of different variables, such as familiarity of participants to each other, and expertise with the technology. Participants were allocated randomly to each condition, so these extraneous variables were already controlled for statistically speaking [36, p. 274]. However, it would be interesting to work out what the effect of these variables would be, e.g. by using them as independent variables, and allocating participants accordingly. Previous studies have looked at the impact of participant's familiarity with others in immersive settings, [110], and although their findings were that mutual history didn't have an impact on participants behaviour, it would be interesting to see if this is the case for larger numbers of participants (e.g. ten participants, like in the desktop studies reported in this thesis).

7.1.2 Virtual time

Chapter 6 took a first step into researching Virtual Time (VT), an asynchronous collaboration method that provides new possibilities for CVE interaction. There are two major research questions remaining for the study of VT: (1) How does it scale? (2) How does the user interact?

The following sections elaborate on these questions, along with possible solutions and ideas for future experimentation.

7.1.2.1 How does it scale?

Imagine a scenario with lots of users interacting in a large-scale CVE over an extended period of time, listening to conversation tags and contributing their own. The possible methods of contributing/recording a conversation tag are discussed in Section 7.1.2.2. There are also fundamental scalability issues from a system's perspective and a human's perspective.

System's perspective The system would need sufficient storage for the audio conversations that are referred to by the tags, and enough bandwidth to transmit the conversations and movements to the real-time users that are present. There are then two possibilities for the interconnectivity of conversation tags.

First, tags could be hierarchical. That is, if a new group of participants made a conversation tag in which they listened to a conversation from the past (and presumably contributed to it in some way), then the playback of the original tag would be nested within the new tag. The tags could be organised in a tree-like structure equivalent to that of Usenet news postings. A tree-like structure would mean the new tag would be a child of the original tag, in the same way a reply to a news posting would be shown as a sub-post beneath the original (and may quote the original within it). Future participants could choose to reply to either the original tag or the child tag.

Alternatively, a flat structure could be imposed by the system. This would mean new participants could extend existing conversation tags, by chipping in and adding comments of their own. However, this would not be stored as a hierarchy. Instead, the amount of conversation contained within each tag would get longer over time.

There are many unanswered questions here. What if two tags are played at once? What problems might be caused by malicious users (e.g. spam, extending tags with irrelevant conversation) and how do these relate to problems with existing asynchronous systems (e.g. going 'off-topic' in an online forum)?

Human's perspective When a human participant enters a VT-enabled environment, they will have a limited time to perform their task and will be presented with a potentially large number of conversation tags containing activity that might take a long time to play back in full. A good design of system will be required to ensure that these tags become a help rather than a hindrance.

First, there is the question of how many users are interacting in real-time? A large number of users could lead to a conflict of interest (two people want to listen to different tags at the same time), but richer conversation being contributed to the environment. Perhaps tag playback could be local to each user to remove conflict, or perhaps users could switch between a private and shared view of the environment.

Second, there is a decision making problem. The user is surrounded by conversation tags: where do they start? The existing system provides a menu based filter that allows each topic of conversation to be hidden or made visible. Additional solutions would be to use techniques from search engines (allow users to search for keywords), or recommender systems. For example, Amazon's 'customers who bought *X* also bought *Y*' could become 'participants who liked conversation *X* also liked these other related conversations: *Y* and *Z*'. How is the recommendation represented in a CVE? How do participants navigate through the recommendations and try different ones?

An alternative would be to provide cues to help people navigate the information rich environment. This would draw on existing spatial cognition work, to determine the usefulness of different cues, e.g. signs vs. maps [53], and maps vs. verbal instructions [71].

7.1.2.2 How does the user interact?

New conversation tags in a VT system can be created, and existing conversation tags can be played back.

In the current implementation of VT, the conversation tags were created manually from the conversations and movements of previous participants in the environment. Keywords were used to determine which conversations to tag (only task-related ones) and the keywords also defined the topic of the tag. In theory, this could have been done using NLP techniques. However, an alternative would be to let users create the tags themselves.

For example, suppose a group of users have found something of interest in the environment. They could explicitly decide to leave a tag about it, and by pressing a 'record' button they could talk about, and look at, the point of interest. The tag would remain for future participants, who could explicitly decide to add to it (the system designer could allow appending or inserting conversation, or a new tag could become a child of the original tag to make a hierarchical structure, see Section 7.1.2.1), or they could make their own new tag. One problem with this would be the potential lack of contributions from participants. What would motivate them to create conversation tags?

Principles from CSCW and organisational psychology could be applied to encourage contributions to the system. For example, recognising the effort of each user and their contributions, [58]. Also, a new task could be chosen to provide an objective measure of performance, and motivation to contribute conversation tags.

An example of a potential task would be virtual tourism, where participants could leave comments about the different places that they have visited. The VT system would record the spatial positioning of participants to put their comments into context. For example, if someone wanted to share their favourite must-see place, or the best restaurant hidden away in the maze of a city, they could lead people there in a virtual representation of the world, show visitors around, leave verbal (audio) comments, and leave a trail for future participants. A task could be set up as a virtual 'treasure hunt', where participants must use the VT system to track down certain information from the environment. People could be divided into teams who must work asynchronously, i.e. they are not all present at the same time and need to leave clues for each other in the form of conversation tags. A reward could be given as a task incentive (e.g. a prize for the best team).

The main objective of the research would be to draw out the principles that are important for collaborative interaction. The same task could be performed in three types of conditions, the latter two being most suited to research. First, to a certain extent, the task could be performed in the real world (e.g. walking around the real historic town or place of interest). However, this would limit the possibilities of asynchronous collaboration (the VT system requires a display unit to show the movements of previous participants). Second, the task could be carried out in a tracking lab (e.g. using one of the motion tracking systems outlined in Section 2.2.1), with a head-mounted display to show the asynchronous VT information—it could show the avatars of previous participants and their conversations (using audio playback). Third, the task could be performed in a desktop CVE, using standard peripherals (mouse, keyboard) for interaction.

A comparison between the second and third conditions could be made: high fidelity movement (tracked) vs. low fidelity movement (desktop). Research has been done by Ruddle and Lessels comparing movement fidelity with respect to navigation [87]. The focus of the VT research would be on collaboration, not navigation, but the metrics provided by Ruddle and Lessels would provide a good foundation for the evaluation [88]. Level 2 and 3 metrics would be most relevant, which are participants' behaviour, and participants' rationale, respectively.

Do people behave differently when they are physically moving around the space, compared to when they are moving in a conventional CVE? The 'joint action' work carried out by Streuber and Chatziastros made a first step in analysing this [112]. Taken as a proof of concept, their study showed that a detailed analysis of the movement and behaviour of participants can be carried out using the motion tracking data. An experimental comparison could be made using a log from a desktop CVE. Would people perform the task better when they were physically moving in the space? If so, could functionality be implemented to replace the benefits of high fidelity movement, so that interaction methods are different (e.g. non-naturalistic), but collaboration is equally effective, or more effective? If interaction methods could be improved it would allow effective collaboration to extend to ubiquitous desktop technologies. Finally, there is the hybrid approach: what if the new interaction methods were combined with the high fidelity movement? What if the non-naturalistic approach was taken to the tracking lab?

The explosion of use of 3D worlds like Second Life and the technical capacity of Internet infrastructure means that all the above are research challenges for today.

Bibliography

[1] IEEE Std 1278. IEEE standard for information technology—protocols for distributed interactive simulation applications: Entity information and interaction, 1993.

[2] Active Worlds. `http://www.activeworlds.com/`. (Accessed 27 October 2008).

[3] AMT3D. `http://www.amt3d.com/3dtechnology.php`. (Accessed 27 April 2007).

[4] A. J. Aretz. The design of electronic map displays. *Human Factors*, 33(1):85–101, 1991.

[5] Ascension Technology Corporation. `http://www.ascension-tech.com/`. (Accessed 2 March 2009).

[6] J. N. Bailenson, N. Yee, J. Blascovich, and R. E. Guadagno. Transformed social interaction in mediated interpersonal communication. In E. A. Konijn, S. Utz, M. Tanis, and S. B. Barnes, editors, *Mediated Interpersonal Communication*, pages 77–99. Routledge, New York, NY, 2008.

[7] K. Baker, S. Greenberg, and C. Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, pages 96–105. ACM, 2002.

[8] B. Becker and G. Mark. Social conventions in computer-mediated communication: A comparison of three online shared virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 19–39. Springer, London, 2002.

[9] S. Benford, J. Bowers, L. E. Fahln, C. Greenhalgh, and D. Snowdon. User embodiment in collaborative virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, pages 242–249. ACM, 1995.

[10] S. Benford, M. Fraser, G. Reynard, B. Koleva, and A. Drozd. Staging and evaluating public performances as an approach to CVE research. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 80–87. ACM, 2002.

[11] J. Blascovich. Social influence within immersive virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 127–145. Springer, London, 2002.

[12] J. Blascovich. A theoretical model of social influence for increasing the utility of collaborative virtual environments. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 25–30. ACM, 2002.

[13] Blizzard Entertainment. Press release. `http://www.blizzard.com/us/press/080122.html`, 2008. (Accessed 22 October 2008).

[14] Blizzard Entertainment. World of Warcraft. `http://www.worldofwarcraft.com/`. (Accessed 22 October 2008).

[15] C. A. Bowers, F. Jentsch, E. Salas, and C. C. Braun. Analyzing communication sequences for team training needs assessment. *Human Factors*, 40(4):672–679, 1998.

[16] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Boston, MA, 2004.

[17] K. Brodlie, D. Duce, J. Gallop, M. Sagar, J. Walton, and J. Wood. Visualization in grid computing environments. In *Proceedings of the IEEE Visualization 2004 Conference*, pages 155–162. IEEE, 2004.

[18] B. Brown and M. Bell. CSCW at play: 'There' as a collaborative virtual environment. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*, pages 350–359. ACM, 2004.

[19] CABE Education and Oxford Brookes University. Making better places—walkabout analysis checklist. `http://www.makingbetterplaces.org.uk/pdfs/analysis_checklist.pdf`, 2004. (Accessed 6 March 2009).

[20] C. Carlsson and O. Hagsand. DIVE—a multi-user virtual reality system. In *Virtual Reality Annual International Symposium*, pages 394–400. IEEE, 1993.

[21] M. Carrozzino, F. Tecchia, S. Bacinelli, C. Cappelletti, and M. Bergamasco. Lowering the development time of multimodal interactive application: the real-life experience of the XVR project. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE '05)*, pages 270–273. ACM, 2005.

[22] L. Cheng, S. Farnham, and L. Stone. Lessons learned: Building and deploying shared virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 90–111. Springer, London, 2002.

[23] LT. Cheng, C. R. B. de Souza, S. Hupfer, J. Patterson, and S. Ross. Building collaboration into IDEs. *ACM Queue*, 1(9):40–50, 2003.

[24] L. Chittaro and S. Burigat. 3D location pointing as a navigational aid for virtual environments. In *Proceedings of the 7th International Conference on Advanced Visual Interfaces (AVI '04)*, pages 267–274. ACM, 2004.

[25] M. Cohen, K. W. Brodlie, and N. Phillips. The volume in focus: Hardware-assisted focus and context effects for volume visualization. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 1231–1235. ACM, 2008.

[26] Commission for Architecture and the Built Environment. The cost of bad design. `http://www.cabe.org.uk/AssetLibrary/8125.pdf`, 2006. (Accessed 27 April 2007).

[27] M. Czerwinski, D. S. Tan, and G. G. Robertson. Women take a wider view. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, pages 195–202. ACM, 2002.

[28] T. DeMarco and T. Lister. *Peopleware: Productive Projects and Teams*. Dorset House Publishing, New York, NY, second edition, 1999.

[29] Deparment of Environment, Transport and the Regions, and Commission for Architecture and the Built Environment. By design: Urban design in the planning system—towards better practice. `http://www.communities.gov.uk/publications/planningandbuilding/bydesignurban`, 2000. (Accessed 17 December 2008).

[30] G. Drettakis, M. Roussou, A. Reche, and N. Tsingos. Design and evaluation of a real-world virtual environment for architecture and urban planning. *Presence: Teleoperators and Virtual Environments*, 16(3):318–332, 2007.

[31] J. Dyck, C. Gutwin, T. C. N. Graham, and D. Pinelle. Beyond the LAN: Techniques from network games for improving groupware performance. In *Proceedings of the 2007 International ACM conference on Supporting Group Work (GROUP '07)*, pages 291–300. ACM, 2007.

[32] Eclipse Foundation. `http://www.eclipse.org/`. (Accessed 18 February 2009).

[33] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[34] Facebook. `http://www.facebook.com/`. (Accessed 27 October 2008).

[35] Fakespace Labs, Inc. `http://www.fakespacelabs.com/`. (Accessed 24 October 2008).

[36] A. Field. *Discovering Statistics Using SPSS*. Sage Publications, 2005.

[37] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. The Systems Programming Series. Addison-Wesley, Reading, MA, second edition, 1996.

[38] B. A. Forouzan. *Data Communications and Networking*. McGraw-Hill, New York, NY, fourth edition, 2007.

[39] M. Fraser, T. Glover, I. Vaghi, S. Benford, and C. Greenhalgh. Revealing the realities of collaborative virtual reality. In *Proceedings of the 3rd International Conference on Collaborative Virtual Environments (CVE '00)*, pages 29–37. ACM, 2000.

[40] E. Frécon. A survey of CVE technologies and systems. *SICS Technical Report*, T2004:03, Swedish Institute of Computer Science, 2004. ISSN 1100-3154.

[41] E. Frécon and M. Stenius. DIVE: a scaleable network architecture for distributed virtual environments. *Distributed Systems Engineering*, 5(3):91–100, 1998.

[42] A. Frisoli, F. Rocchi, S. Marcheschi, A. Dettori, F. Salsedo, and M. Bergamasco. A new force-feedback arm exoskeleton for haptic interaction in virtual environments. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (World Haptics '05)*, pages 195–201. IEEE, 2005.

[43] C. Greenhalgh and S. Benford. MASSIVE: A collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction*, 2(3):239–261, 1995.

[44] C. Greenhalgh and S. Benford. Supporting rich and dynamic communication in large-scale collaborative virtual environments. *Presence: Teleoperators and Virtual Environments*, 8(1):14–35, 1999.

[45] C. Greenhalgh, M. Flintham, J. Purbrick, and S. Benford. Applications of temporal links: Recording and replaying virtual environments. In *Proceedings of IEEE Virtual Reality (VR '02)*, pages 101–108. IEEE, 2002.

[46] C. Greenhalgh, J. Purbrick, and D. Snowdon. Inside MASSIVE-3: flexible support for data consistency and world structuring. In *Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE '00)*, pages 119–127. ACM, 2000.

[47] S. Harrison and P. Dourish. Re-place-ing space: The roles of place and space in collaborative systems. In *Proceedings of the 1996 ACM conference on Computer Supported Cooperative Work (CSCW '96)*, pages 67–76. ACM, 1996.

[48] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*. Pearson Prentice Hall, NJ, third edition, 2004.

[49] I. Heldal, M. Spante, and M. Connell. Are two heads better than one? Object-focussed work in physical and in virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '06)*, pages 287–296. ACM, 2006.

[50] G. Hiebert. OpenAL 1.1 Specification and Reference. `http://connect.creativelabs.com/openal/Documentation/OpenAL%201.1%20Specification.pdf`, 2005. (Accessed 14 October 2008).

[51] J. Hindmarsh, M. Fraser, C. Heath, S. Benford, and C. Greenhalgh. Object-focused interaction in collaborative virtual environments. *ACM Transactions on Computer-Human Interaction*, 7(4):477–509, 2000.

[52] J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*, 7(2):174–196, 2000.

[53] C. Hölscher, S. J. Büchner, M. Brösamle, T. Meilinger, and G. Strube. Signs and maps—cognitive economy in the use of external aids for indoor navigation. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society (CogSci '07)*, pages 377–382. Cognitive Science Society, 2007.

[54] IBM Rational Jazz. `http://jazz.net`. (Accessed 18 February 2009).

[55] K. Jää-Aro and D. Snowdon. How not to be objective. In E. F. Churchill, D. N. Snowdon, and A. J. Munro, editors, *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*, pages 143–159. Springer, London, 2001.

[56] R. Johansen. *GroupWare: Computer Support for Business Teams*. The Free Press, New York, NY, 1988.

[57] N. P. Juster, J. Maxfield, P. M. Dew, S. Taylor, M. Fitchie, W. J. Ion, J. Zhao, and M. Thompson. Predicting product aesthetic quality using virtual environments. *Journal of Computing and Information Science in Engineering*, 1(2):105–112, 2001.

[58] N. L. Kerr and R. S. Tindale. Group performance and decision making. *Annual Review of Psychology*, 55:623–655, 2004.

[59] J. Kotlarsky and I. Oshri. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, 14(1):37–48, 2005.

[60] T. Koyama, I. Yamano, K. Takemura, and T. Maeno. Multi-fingered exoskeleton haptic device using passive force feedback for dexterous teleoperation. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2905–2910. IEEE, 2002.

[61] R. E. Kraut, S. R. Fussell, and J. Siegel. Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction*, 18(1):13–49, 2003.

[62] J. Lai, A. Levas, P. Chou, C. Pinhanez, and M. Viveros. Bluespace: personalizing workspace through awareness and adaptability. *International Journal of Human-Computer Studies*, 57(5):415–428, 2002.

[63] S. Lessels and R. A. Ruddle. Changes in navigational behaviour produced by a wide field of view and a high fidelity visual scene. In *Proceedings of the 10th Eurographics Symposium on Virtual Environments (EGVE'04)*, pages 71–78. The Eurographics Association, 2004.

[64] Kurt Lewin. *Resolving social conflicts : selected papers on group dynamics*. Research Center for Group Dynamics, University of Michigan. Harper & Brothers, New York, NY, 1948.

[65] Linden Lab. Second Life. `http://www.secondlife.com/`. (Accessed 24 October 2008).

[66] J. M. Linebarger, C. D. Janneck, and G. D. Kessler. Leaving the world behind: Supporting group collaboration patterns in a shared virtual environment for product design. *Presence: Teleoperators and Virtual Environments*, 14(6):697–719, 2005.

[67] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, and S. Zeswitz. NPSNET: A network software architecture for large scale virtual environments. *Presence: Teleoperators and Virtual Environments*, 3(4):265–287, 1994.

[68] I. MacInnes. Property rights, legal issues, and business models in virtual world communities. *Electronic Commerce Research*, 6(1):39–56, 2006.

[69] J. D. Mackinlay, S. K. Card, and G. G. Robertson. Rapid controlled movement through a virtual 3D workspace. *Computer Graphics*, 24(4):171–176, 1990.

[70] Makena Technologies, Inc. There. `http://www.there.com/`. (Accessed 24 October 2008).

[71] T. Meilinger. Wayfinding with maps and verbal directions. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society (CogSci '05)*. Cognitive Science Society, 2005.

[72] Microsoft Visual Studio Team System. `http://msdn.microsoft.com/en-us/teamsystem/default.aspx`. (Accessed 18 February 2009).

[73] R. J. Moore, N. Ducheneaut, and E. Nickell. Doing virtually nothing: Awareness and accountability in massively multiplayer online worlds. *Computer Supported Cooperative Work (CSCW)*, 16(3):265–305, 2007.

[74] N. Murray and D. Roberts. Comparison of head gaze and head and eye gaze within an immersive environment. In *Proceedings of the Tenth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT '06)*, pages 70–76. IEEE, 2006.

[75] A. Nilsson, I. Heldal, A. Axelsson, and R. Schroeder. The long term uses of shared virtual environments: An exploratory study. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 112–126. Springer, London, 2002.

[76] Nintendo Wii. `http://www.nintendo.com/wii`. (Accessed 2 March 2009).

[77] G. M. Olson and J. S. Olson. Distance matters. *Human-Computer Interaction*, 15(2):139–178, 2000.

[78] X. Pan, M. Gillies, and M. Slater. Male bodily responses during an interaction with a virtual woman. In *Intelligent Virtual Agents*, pages 89–96. Springer, 2008.

[79] S. Pekkola. Critical approach to 3D virtual realities for group work. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction (NordiCHI '02)*, pages 129–138. ACM, 2002.

[80] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 1285–1294. ACM, 2008.

[81] Photosynth. `http://livelabs.com/photosynth`. (Accessed 12 February 2009).

[82] D. Roberts and R. Wolff. Controlling consistency within collaborative virtual environments. In *Proceedings of the Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT '04)*, pages 46–52. IEEE, 2004.

[83] D. Roberts, R. Wolff, O. Otto, and A. Steed. Constructing a gazebo: Supporting teamwork in a tightly coupled, distributed task in virtual reality. *Presence: Teleoperators and Virtual Environments*, 12(6):644–657, 2003.

[84] Robust Audio Tool. `http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/`. (Accessed 7 March 2009).

[85] S. Rohall, LT. Cheng, and J. Patterson. Bluegrass: Lessons for deploying virtual worlds within the enterprise. In *Workshop on Virtual Worlds, Collaboration, and Workplace Productivity, (CSCW '08)*. ACM, 2008.

[86] R. A. Ruddle, A. Howes, S. J. Payne, and D. M. Jones. The effects of hyperlinks on navigation in virtual environments. *International Journal of Human-Computer Studies*, 53(4):551–581, 2000.

[87] R. A. Ruddle and S. Lessels. For efficient navigational search humans require full physical movement but not a rich visual scene. *Psychological Science*, 17(6):460–465, 2006.

[88] R. A. Ruddle and S. Lessels. Three levels of metric for evaluating wayfinding. *Presence: Teleoperators and Virtual Environments*, 15(6):637–654, 2006.

[89] R. A. Ruddle, S. J. Payne, and D. M. Jones. The effects of maps on navigation and search strategies in very-large-scale virtual environments. *Journal of Experimental Psychology: Applied*, 5:54–75, 1999.

[90] R. A. Ruddle, J. C. D. Savage, and D. M. Jones. Verbal communication during cooperative object manipulation. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 120–127. ACM, 2002.

[91] H. Sacks. *Lectures on Conversation*, volume 2. Blackwell, Oxford, 1992.

[92] M. Salanova, S. Llorens, E. Cifre, I. M. Martnez, and W. B. Schaufeli. Perceived collective efficacy, subjective well-being and task performance among electronic work groups. *Small Group Research*, 34(1):43–73, 2003.

[93] E. Sallnäs, K. Rassmus-Gröhn, and C. Sjöström. Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(4):461–476, 2000.

[94] R. Schroeder. Social interaction in virtual environments: Key issues, common themes, and a framework for research. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 1–18. Springer, London, 2002.

[95] R. Schroeder. An overview of ethical and social issues in shared virtual environments. *Futures*, 39(6):704–717, 2007.

[96] R. Schroeder, I. Heldal, and J. Tromp. The usability of collaborative virtual environments and methods for the analysis of interaction. *Presence: Teleoperators and Virtual Environments*, 15(6):655–667, 2006.

[97] R. Schroeder, A. Steed, A. Axelsson, and I. Heldal. Collaborating in networked immersive spaces: as good as being there together? *Computers & Graphics*, 25(5):781–788, 2001.

[98] SensAble Technologies, Inc. `http://www.sensable.com/`. (Accessed 26 February 2009).

[99] S. Singhal and M. Zyda. *Networked virtual environments : design and implementation*. ACM SIGGRAPH and ACM Press and Addison-Wesley, 1999.

[100] Skype Ltd. `http://www.skype.com/`. (Accessed 7 March 2009).

[101] M. Slater, A. Antley, A. Davison, D. Swapp, C. Guger, C. Barker, N. Pistrang, and M. V. Sanchez-Vives. A virtual reprise of the Stanley Milgram obedience experiments. *PLoS ONE*, 1(1):e39, 2006.

[102] M. Slater, A. Sadagic, M. Usoh, and R. Schroeder. Small-group behavior in a virtual and real environment. *Presence: Teleoperators and Virtual Environments*, 9(1):37–51, 2000.

[103] M. Slater and A. Steed. Meeting people virtually: Experiments in shared virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 146–171. Springer, London, 2002.

[104] R. B. Smith, R. Hixon, and B. Horan. Supporting flexible roles in a shared space. In E. F. Churchill, D. N. Snowdon, and A. J. Munro, editors, *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*, pages 160–176. Springer, London, 2001.

[105] D. H. Sonnenwald, M. C. Whitton, and K. L. Maglaughlin. Evaluating a scientific collaboratory: Results of a controlled experiment. *ACM Transactions on Computer-Human Interaction*, 10(2):150–176, 2003.

[106] Splash Damage Ltd. Wolfenstein: Enemy Territory. `http://www.splashdamage.com/`. (Accessed 27 October 2008).

[107] SR Research Ltd. `http://www.sr-research.com/`. (Accessed 2 March 2009).

[108] K. M. Stanney. *Handbook of virtual environments : design, implementation, and applications*. Lawrence Erlbaum Associates, Mahwah, NJ, 2002.

[109] A. Steed, M. Slater, A. Sadagic, A. Bullock, and J. Tromp. Leadership and collaboration in shared virtual environments. In *Proceedings of IEEE Virtual Reality (VR '99)*, pages 112–115. IEEE, 1999.

[110] A. Steed, M. Spante, I. Heldal, AS Axelsson, and R. Schroeder. Strangers and friends in caves: an exploratory study of collaboration in networked IPT systems for extended periods of time. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 51–54. ACM, 2003.

[111] R. J. Stone. Haptic feedback: a brief history from telepresence to virtual reality. In S. Brewster and R. Murray-Smith, editors, *Haptic Human-Computer Interaction*, pages 1–16. Springer Berlin / Heidelberg, 2001.

[112] S. Streuber and A. Chatziastros. Human interaction in multi-user virtual reality. In *Proceedings of the 10th International Conference on Humans and Computers (HC '07)*, pages 1–6, 2007.

[113] J. Suleiman and R. T. Watson. Social loafing in technology-supported teams. *Computer Supported Cooperative Work (CSCW)*, 17(4):291–309, 2008.

[114] S. Teasley, L. Covi, M. S. Krishnan, and J. S. Olson. How does radical collocation help a team succeed? In *Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work (CSCW '00)*, pages 339–346. ACM, 2000.

[115] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics*, 23(3):249–258, 2004.

[116] B. W. Tuckman. Developmental sequence in small groups. *Psychological Bulletin*, 63(6):384–399, 1965.

[117] Vicon. `http://www.vicon.com/`. (Accessed 11 December 2008).

[118] D. L. Weatherford. Representing and manipulating spatial information from different environments: Models to neighborhoods. In R. Cohen, editor, *The Development of Spatial Cognition*, pages 41–70. Lawrence Erlbaum Associates, Florence, Kentucky, 1985.

[119] WorldViz. `http://www.worldviz.com/`. (Accessed 24 October 2008).

[120] U. Wössner, J. P. Schulze, S. P. Walz, and U. Lang. Evaluation of a collaborative volume rendering application in a distributed virtual environment. In *Proceedings of the workshop on Virtual Environments 2002 (EGVE '02)*, pages 113–221. The Eurographics Association, 2002.

[121] Xsens Technologies. `http://www.xsens.com/`. (Accessed 2 March 2009).

[122] H. Yang and G. M. Olson. Exploring collaborative navigation: the effect of perspectives on group performance. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 135–142. ACM, 2002.

[123] N. Yee. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: Teleoperators and Virtual Environments*, 15(3):309–329, 2006.

[124] N. Yee and J. N. Bailenson. The proteus effect: The effect of transformed self-representation on behavior. *Human Communication Research*, 33(3):271–290, 2007.

[125] N. Yee, J. N. Bailenson, and N. Ducheneaut. The proteus effect: Implications of transformed digital self-representation on online and offline behavior. *Communication Research*, 36(2):285–312, 2008.

[126] N. Yee, J. N. Bailenson, M. Urbanek, F. Chang, and D. Merget. The unbearable likeness of being digital: The persistence of nonverbal social norms in online virtual environments. *Cyberpsychology & Behavior*, 10(1):115–121, 2007.

[127] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. *SIGCHI Bulletin*, 17(SI):189–192, 1987.

Appendix A

Urban planning guidelines

The following information sheet was provided to all participants of the experiments, with references to mobile group dynamics and virtual time removed for participants in conditions where that functionality was not available.

Virtual Environment Study

The purpose of this study is to investigate the use of 'mobile group dynamics' and 'virtual time' to improve upon traditional collaborative interaction in virtual environments. The environment is a 3D representation of a residential estate. Users should work together, discuss the points outlined below (adapted from guidelines in documents[1,2]) and fill in their own 'urban planning report' sheet.

Guideline 1 – 'Permeability' – An accessible environment with a choice of routes

Consider first the development as a whole. Identify entry and exit points to the estate, and routes around the environment.

- How many entrance and exit points are there around the estate? What are these for (i.e. cars or pedestrians)?
- Discuss ease of movement for cars and pedestrians. What reduces the speed/volume of traffic? Are there suitable pedestrian routes around the environment?
- Consider the block size. This should be small enough to create a variety of routes around the environment and make it permeable. Are the blocks small enough or do you have to walk too far before you reach a choice of direction?

Guideline 2 – 'Character' – A place with its own identity

An environment should provide a sense of place. For example, some parts of cities have landmarks and buildings of civic importance, or natural features (rivers and canals). In the absence of these, the character of an environment is defined by appearance (e.g. the size of buildings and the choice of building materials) and layout.

- Which parts of the environment follow the same pattern/building structure?
- Find a part of the environment that is not consistent with the layout of the estate. Is this acceptable, or should it be changed?
- Does the estate have character?

Guideline 3 – 'Safety and security' – A place where public places are overlooked and private spaces are enclosed

Read the following extracts from urban planning literature:

> 'Buildings with live edges, such as shopfronts, doors directly to the street, or residential upper floors, enable people to keep an eye on public space and make it feel safer.'
>
> 'Gaps between buildings reduce the degree to which the street is overlooked, as do blank walls (which also encourage graffiti).'
>
> 'Clearly defining and enclosing private space at the back of buildings provides for better privacy and security. Back yards or inner courtyards that are private or communally shared space are best enclosed by the backs of buildings. The rear gardens of houses are more secure if they back on to other gardens, rather than side roads, service lanes or footpaths.' – (ODPM, 2000)

- Discuss the safety and security of the estate based on your own thoughts and the information above. Find examples of where public and private space is clearly distinguished and where it isn't.
- Discuss which part(s) of the estate you think are least safe. Can you find any blank walls that you think should be overlooked to improve the feeling of safety and help prevent graffiti?
- Discuss any improvements you think could be made with regard to the safety and security of the estate.

---

[1] CABE Education and Oxford Brookes University. Making better places - walkabout analysis checklist. http://www.makingbetterplaces.org.uk/, 2004. (Accessed 21/8/2006).

[2] ODPM. Urban design in the planning system: Towards better practice. http://www.odpm.gov. uk/index.asp?id=1145239, 2000. (Accessed 8/2/2006).

Appendix B

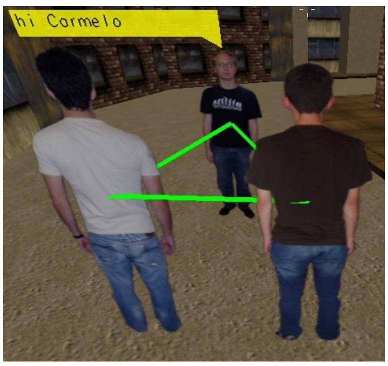
User interface controls

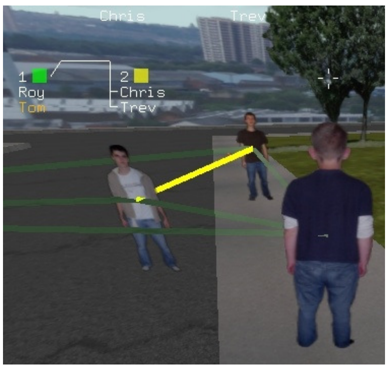
The following sheets were provided to the participants for the basic MGDs condition and multiple views conditions respectively. Other conditions had similar instructions with the relevant modifications.

How to use the urban planning review system

- **Movement** – Move using the **arrow keys**. Look around using the **mouse**. Raise/lower your viewpoint with the **Home** key.
- **Group movement** – **Right-click** on people in your group to **follow** them. Move (press an arrow key) to stop following them. Press the **middle mouse button** to automatically move to your group's mean location.
- **Communication** – Hold **Page Down** to talk using the mic. **Type some text** and press return to talk using text.
- **Screenshot** – Press **Insert** to take a **screenshot**.

You can hear your group/subgroup members wherever you are in the environment. Other people will get quieter as they walk away.

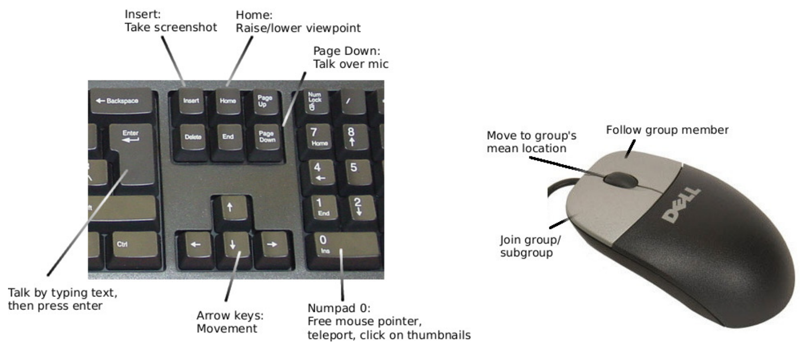




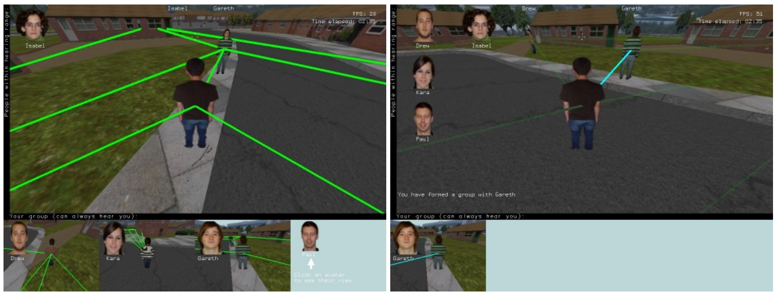
Groups are shown with coloured lines between members (left screenshot). Subgroups are shown a different colour, with the parent group semi-transparent (right screenshot). The group/subgroup makeup is shown in the corner of the screen, showing the group number, its colour, and a list of all the group's members (right screenshot).

How to use the urban planning review system

- **Movement** – Move using the **arrow keys**. Look around using the **mouse**. Raise/lower your viewpoint with the **Home** key.
- **Group movement** – **Right-click** on people in your group to **follow** them. Move (press an arrow key) to stop following them. Press the **middle mouse button** to automatically move to your group's mean location.
- **Communication** – Hold **Page Down** to talk using the mic. **Type some text** and press return to talk using text.
- **Screenshot** – Press **Insert** to take a **screenshot**.
- **Teleport** – Hold **Numpad 0** to free the mouse pointer. Position the teleport destination by moving the mouse, **left-click** to teleport. Click a **thumbnail view** of a group member to move to their location.



You can hear your group members wherever you are in the environment. You will hear other users in 3D (i.e. stereo sound getting quieter as they move further away, until they reach a cut-off point). People within hearing range are listed down the left-hand side of the screen.



Groups are shown with coloured lines between members (left screenshot). Subgroups are shown a different colour, with the parent group semi-transparent (right screenshot).

You can see thumbnails of your group member's views across the bottom of the screen. A maximum of three thumbnails will be shown. Extra group members are shown in the bottom-right section of the screen (left screenshot). Click on them with the mouse (by holding down Numpad 0) to swap them into a thumbnail space.

Appendix C

Publications

The following publications are reproduced here.

T. J. Dodds and R. A. Ruddle. Mobile group dynamics in large-scale collaborative virtual environments. In *Proceedings of IEEE Virtual Reality (VR '08)*, pages 59–66. IEEE Press, 2008.

T. J. Dodds and R. A. Ruddle. Using teleporting, awareness and multiple views to improve teamwork in collaborative virtual environments. In *Proceedings of the 14th Eurographics Symposium on Virtual Environments (EGVE '08)*, pages 81–88. The Eurographics Association, 2008.

T. J. Dodds and R. A. Ruddle. Using Mobile Group Dynamics and Virtual Time to improve teamwork in large-scale Collaborative Virtual Environments. *Computers and Graphics*, 33(2):130–138, 2009.

Mobile Group Dynamics 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.

ABSTRACT

We have developed techniques called Mobile Group Dynamics (MGDs), which help groups of people to work together while they travel around large-scale virtual environments. MGDs explicitly showed the groups that people had formed themselves into, and helped people move around together and communicate over extended distances. The techniques were evaluated in the context of an urban planning application, by providing one batch of participants with MGDs and another with an interface based on conventional collaborative virtual environments (CVEs). Participants with MGDs spent nearly twice as much time in close proximity (within 10m of their nearest neighbor), communicated seven times more than participants with a conventional interface, and exhibited real-world patterns of behavior such as staying together over an extended period of time and regrouping after periods of separation. The study has implications for CVE designers, because it shows how MGDs improves groupwork in CVEs.

Keywords: Collaborative interaction, experimental methods, distributed VR, usability

Index Terms: C.2.4 [Computer-Computer Communication Networks]: Distributed Systems—Distributed applications; H.1.2 [Models and Principles]: User/Machine Systems—Human factors; Software psychology; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces—Collaborative computing; Computer-supported cooperative work; Synchronous interaction; I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Virtual Reality

1 INTRODUCTION

Collaborative virtual environments (CVEs) are three dimensional electronic worlds that combine shared information (e.g. 3D design models) with mechanisms that allow multiple people to co-exist, be aware of each other's presence (e.g. through avatars) and communicate. CVEs are used for games and social communication, but more general usage is inhibited by current mechanisms for collaborative interaction.

Our goal is to allow people to interact in CVEs as effectively as they do in the real world. We aim to achieve this by developing techniques that support 'group dynamics' (the processes by which people form themselves into groups and operate), as people travel around (i.e. are mobile) and work together in a large-scale shared space. Large-scale spaces are those in which 'Multiple vantage points must be occupied in order for the space to be visually apprehended in its entirety.' [20] (p. 42). This introduces extra challenges, because not only do individuals get easily disoriented when they navigate a large-scale VE, it is also all too easy to lose track of the whereabouts of one's collaborators.

This paper describes the implementation of mobile group dynamics (MGDs), and its evaluation using an urban planning scenario in which one group of participants were provided with MGD functionality and another ('control') group were not. First, however, aspects of group dynamics that we often take for granted in the real world are reviewed along with methods used to support group work in both publicly accessible CVEs and research applications.

2 BACKGROUND

The field of group dynamics has long been studied within a socially driven context in real life (e.g. [12]), and the much-cited model of *forming*, *storming*, *norming* and *performing* has been constructed to describe group processes that are involved [19]. Storming and norming are the processes by which individuals' roles within a group become refined, whereas forming and performing govern the creation of groups and their ability to do work. It is these latter two processes that are most relevant to implementing MGDs.

2.1 Forming

Four key points about group formation need to be considered. These are the method of joining (implicit vs. explicit), how members are identified, the structure of the group (e.g. subgroups/hierarchy), and the way that the group is represented (e.g. aggregate views of the group as a whole).

When people meet and communicate informally in the real world they gather together into circles to hear each other. The groups are organized using spatial positioning so membership is implicit, and social etiquette applies when people join or leave. For example, new members may be invited to join by existing members' body language (e.g. stepping back to allow a newcomer into the circle), and when members leave the group they would often give an appropriate verbal indication or gesture (e.g. say or wave goodbye).

Active Worlds[1] is a chat-based CVE in which users form implicit groups. If users are too far apart, the chat text isn't displayed, so they are forced to gather together into rough circles to 'hear' each other. Groups can make themselves open to new members by gathering around the entrances to the worlds, or groups can govern themselves by agreeing a time and place to meet. The environments are large enough for this to provide privacy from users who were not invited because a group is unlikely to be found by accident. However, a disadvantage of this implicit approach is that the system maintains no record of the makeup of each group, so members may be unaware if they met by chance in another part of the CVE.

On the other hand, people can be part of explicit groups for example a guest list for a wedding, a university society or sports club. Explicit groups maintain a formal record of their membership. In some cases membership is open (any student can join a society, they just need to sign up) but in others it is dictated by members

*e-mail: trev@comp.leeds.ac.uk
†e-mail: royr@comp.leeds.ac.uk

[1] http://www.activeworlds.com/ (Accessed 2 August 2007)

who have special privileges (e.g., a couple deciding who they will invite to their wedding).

Social networking sites such as Facebook[2] use explicit groups. Membership can be decided by a group administrator (as in the guest list example) or it can be open to anyone (as in a society or club). *There*[3] is a chat-based CVE that implements explicit groups in a similar way to social networking sites. It uses a web based interface to allow the forming and joining of groups for people with a particular common interest. However, there is no way of identifying who belongs to one of these groups from the 3D environment, or from the appearance of users' avatars. The only way to identify members is by consulting the group membership lists.

Group members can also be identified by spatial positioning or color schemes. The former approach is used by real-time chat groups in There. A group is started after one user has chatted with another for a short period of time and the camera automatically switches to a special 'chat mode' that shows the users' two avatars aligned side-by-side, giving a better view of the group. New members can join by walking up to the group, clicking on an icon associated with it, and selecting the join option from a menu that appears. As more people join the avatars are arranged into a semi-circle, so each user can see all the group members in one view on screen. The disadvantage of this approach is it only works when the group remains in one location.

A soccer team provides a good example of the real-world use of color schemes. Membership is decided explicitly before the match starts, and it is communicated by the players wearing their team's colors. There is no question who is on which team, and it is straightforward to identify who is a member from a distance. A similar approach is used in entertainment CVEs such as Wolfenstein: Enemy Territory[4] where members of the two opposing armies can be identified from the uniform worn by users' avatars.

Even if there is only one group in the environment, it may change structure. For example, consider an office meeting. The people present may divide themselves into subgroups to carry out certain tasks, or someone may talk to the person next to them, using 'side-channels' of communication [3] rather than addressing the group as a whole. Functionality to support changes to a group's structure have rarely been implemented in CVEs, but a recent exception was [13], who allowed users to form subgroups explicitly using menu-based selection.

Finally, the way a group is represented can change. This is specific to CVEs. For example, MASSIVE-2 [9] implemented a concept of 'third party objects' – objects that affect the awareness between other objects (i.e. users). In their 'Arena' work, they used third party objects to hold crowds of users. Members of a crowd could see other individual members, but non-members saw an aggregate view instead (a large avatar).

2.2 Performing

Performing is the stage in which the group carry out the task. When a group of people work together in a large-scale space in the real world, they communicate and move around the environment. Communication can take place when members are collocated, or when they are physically separated (e.g. communicating using a mobile phone). The Robust Audio Tool (RAT) used in the COVEN project is an example of an audio system that is independent of the spatial positioning of users in a CVE [17]. All users hear each other at all times, as in a typical audio conference. Other CVEs use much more realistic audio, such as the binaural sound system used by Tsingos et al. [18] in which a user wearing headphones can pinpoint the source of each sound. The advantage of RAT style audio is that users can continue communicating wherever they are in the environment. However, this type of audio doesn't scale well because of the noise of users talking over each other. The advantages of an environment using 3D audio include helping the user comprehend who is talking (because one can mentally map the source of the sound to the visual avatar), and reducing noise from multiple users by culling the distant sound sources, so that listeners only hear their neighbors.

Movement around an environment in the real world can be individual (people split up and divide the task between them), as a group (to get a shared understanding), or require meeting at a point of interest [21]. Moving as individuals or as a group both have their advantages. Dividing the environment up between group members is a quicker way of covering the space, but navigating together allows the sharing of ideas – 'two heads are better than one' – and mistakes in the task are less likely to go unnoticed. The group can take a hybrid approach and divide into subgroups, increasing speed of task performance and still benefiting from a small amount of groupwork.

Moving as a group in a virtual world is a non-trivial task, due to the small field of view in desktop environments (it is easy to lose track of where other users are). An over-the-shoulder perspective helps when compared to a first-person perspective: users can see others relative to their avatar [6]. However, moving the camera behind the avatar just provides a bit more context, not a larger field of view, and difficulties still occur [11]. One solution is to use an abstract device to provide an indication of where others are (e.g. a radar, or 3D arrows pointing to targets [7]).

3 Implementing Mobile Group Dynamics

The MGD techniques were designed to make it easier for groups to form in the CVE, and to support their operation as they performed the task. They differed from prior work by using a novel 'group graph' metaphor for users to keep track of each other (Section 3.1) and an easy mechanism for switching between moving as individuals vs. a group (Section 3.2). We describe the techniques here, and Section 4.1.4 gives full details of the interface controls.

3.1 Forming

Forming or joining a group could be done implicitly or explicitly, under one of the following conditions:

- **Implicit:** Moving within 1m of another participant's avatar.
- **Explicit:** Selecting another participant's avatar.

A new group was formed if neither participant was already in a group. The group was joined if one participant was not in a group and the other was. If both participants were in different groups, then the implicit condition had no effect. For the explicit condition, one selection would move the participant out of their current group, and a second selection (or satisfying the implicit condition) was required to move them into the other's group.

The group system was hierarchical, and this worked with explicit selection only. A subgroup was formed if selection occurred and both participants were already in the same group. Leaving the group happened one step at a time. First participants would be returned to their parent level of the hierarchy if they were in a subgroup, and they would be removed from their group altogether if they were at the top-level.

To help groups function over extended time periods, and encourage group members to get back together again after periods apart, the composition of groups at any given moment was identified explicitly by: (a) a group graph that linked participants with a unique color for each group (see Figure 2), and (b) a list of the participants in each group displayed using a Head-up Display (HUD) metaphor (i.e. a transparent overlay).

[2]http://www.facebook.com (Accessed 1 August 2007)

[3]http://www.there.com (Accessed 1 August 2007)

[4]http://www.splashdamage.com/ (Accessed 2 August 2007)

The lines between avatars in each group graph provided an indication of where others were in the environment (e.g. the location of fellow group members). This was particularly clear in a birds-eye view, when the groups could be seen as independently colored graphs, with the names of participants at the nodes. Delaunay triangulation was used to determine which graph edges should be drawn, thereby reducing clutter in the environment. The consequences of this were that participants didn't necessarily have a line from their avatar to every group member, and the edges changed as participants moved.

3.2 Performing

Techniques were implemented to support groups as they communicated and moved around the environment.

A suite of functionality was provided to assist movement as a group. Participants could automatically follow a group member or move to the mean location of their group, and another benefit of this functionality was that it could be used to rapidly move to a group member's location. During automatic movement participants still had full control of their orientation, so they could look around while being 'taken' somewhere. This meant that they could get an understanding of where they were heading, and continue to look at things in the environment while the automatic movement carried them along. To stop the automatic movement (e.g. to stop following someone), participants simply pressed one of their movement keys.

Collision detection remained enabled during the movement to the mean location of the group because the mean location might have resided in an out of bounds area (e.g. a building). A sliding algorithm smoothly moved participants along walls/fences.

Finally, the communication model provided 3D audio communication (see Section 4.1.2). This had two benefits. Firstly, participants had a clearer indication of the the location of someone who was talking, from the direction of the sound (particularly helpful if their avatar was out of sight). Secondly, for people in the same group, the volume level was not affected by the distance between people, which helped collaboration continue even when they were far apart in the environment. However, distance attenuation was implemented for inter-group communication, to reduce the overall noise levels.

4 Experiment

Participants were asked to review a 3D representation of a residential estate that was presented in a CVE system, and complete an urban planning report. The experiment was carried out in two batches. Participants in the first batch were provided with the MGD functionality that we'd developed to aid collaboration in large-scale VEs (MGDs condition), whereas in the second batch MGDs were disabled so functionality was typical of current CVEs (a 'control' condition).

4.1 Method

The experiment took place in an undergraduate computing laboratory. Each participant used two adjacent computers, one for the CVE, and the other for the write-up of their urban planning report. Participants were spaced out across the laboratory so they could only communicate using audio and text communication from within the environment.

4.1.1 Participants

All participants were undergraduate students from the School of Computing. Ten participants were recruited for each run, but two participants for the MGDs condition were unavailable on the day of the experiment. The remaining eight participants (5 men and 3 women) had a mean age of 20.8 ($SD = 2.0$). The ten participants (9 men and 1 woman) in the control condition (MGDs disabled) had a mean age of 22.0 ($SD = 3.5$).

All the participants volunteered for the experiment, gave informed consent and were paid an honorarium for their participation.

4.1.2 CVE application

The software was written in C++ using OpenGL and OpenAL, programmed by the first author. The system allowed multiple participants to connect simultaneously to the environment, be aware of the position and orientation of each other, and communicate using audio and text mediums.

The software used a client-server architecture, using UDP/IP for voice communication and TCP/IP for all other data (e.g. movement, MGD information, text communication). The method for sending data was based on the Protocol Data Unit (PDU) system from the Distributed Interactive Simulation (DIS) protocol [16]. Each data unit had a header identifying the user that the unit originated from and the type of data being sent, followed by the data itself (e.g. positional data consisted of 5 floating point numbers: x, y and z coordinates followed by heading and pitch). The server recorded all the data to a log file, with timestamps so sessions could be played back at their original speed.

Audio communication was implemented using OpenAL 1.1, using the 'inverse distance clamped' model, a reference distance (refdist) of 30m and a roll-off factor (rolloff) of 6. This means that for distances (dist) of up to 30m the gain was 1.0, between 30 and 85m the gain was defined by the equation $\frac{refdist}{refdist+rolloff(dist-refdist)}$ [10]. This gave a gain of 0.08 at 85m, and beyond this the gain was set to zero.

The stereo channels were used to help participants pinpoint the sound source. If a source was to the right of the listener or central, then the gain of the right stereo channel was kept the same (the gain calculated by the attenuation model). As the source moved to the left of the listener, the right channel gain was reduced from 100% (central) to 0% (directly to the left of the listener), and vice-versa for the left channel. This is calculated by the OpenAL implementation.

Distance attenuation was turned off for communications between members of the same group in the MGDs condition. This helped group members communicate as they traveled to different parts of the estate.

To further help participants identify who was talking, an icon was placed above a participant's avatar when they were talking, as a visual cue.

The experiment took place on a Linux platform across a 100 Mbit/s LAN. However, the system (including audio transmission) was tested and ran successfully on Linux and Windows platforms across the Internet on a home broadband connection (2Mbit/s).

4.1.3 Environment

The environment was a residential estate that was based on a real estate in Leeds. The estate was chosen after a murder took place which highlighted one way in which the estate's design didn't follow present UK urban planning guidelines. It occurred in a private space that was only partially enclosed – it was not separated from a public footpath that ran along side it, and on the other side of the footpath was a public park. This broke the following guideline:

> 'Clearly defining and enclosing private space at the back of buildings provides for better privacy and security.
>
> - Back yards or inner courtyards that are private or communally shared space are best enclosed by the backs of buildings.
> - The rear gardens of houses are more secure if they back on to other gardens, rather than side roads, service lanes or footpaths.' [14]

The incident served as a reminder of the importance of good design. Unfortunately, the pressures for short term financial savings have been known to compromise good design, and mistakes remain for years to come [8].

All participants were represented in the environment with a photographic avatar (using four photos: front, back, left and right, see Figure 3). Participants were given an over-the-shoulder perspective, with the option of switching to and from a bird's-eye view. An over-the-shoulder perspective meant that participants could see each other relative to their avatar, and be more aware of how others perceived them [6].

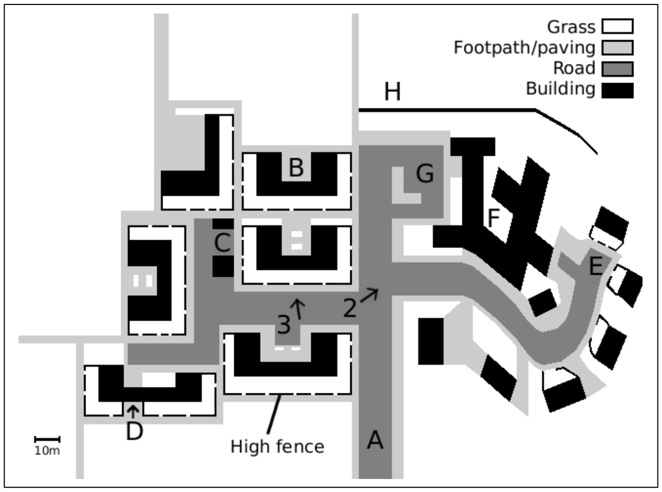


Figure 1: A map of the estate. The estate had an entrance road in the middle (point A), which acted as a dividing line between two styles of building. On the left-hand side of the entrance road, there were brown-bricked terraced houses (see Fig 3), which were mostly horseshoe shapes creating partially enclosed private space (e.g. point B). The front gardens were bordered by high fences, and there were six garages in the road (C). There was an archway under one of the terraces (D). On the right-hand side of the entrance road there were red-bricked bungalows (single story buildings shown in Fig 2) along the edge of the curved road, with gardens bordered by low brick walls (e.g. E). There was a single-story care home for elderly people (F, the large building in Fig 2), with a car park to the left with space for six cars (G), and a hedge-row above it partly separating private land around the care home from public parkland (H). Points 2 & 3 show the position and direction of view used for Figs 2 & 3, respectively.

4.1.4 User Interface

The participants used desktop workstations, and a two-handed control method, with one hand on the keyboard and the other hand on a 3-button mouse. By holding down appropriate arrow keys a participant could move forward/backward/left/right at 6 m/s, and heading and pitch could be changed by moving the mouse. This is a common gaming control method (e.g. [5]).

The 'Insert' key was used to take screenshots, the 'Home' key to toggle between over-the-shoulder and bird's-eye views, and holding down the 'Page Down' key allowed the participant to use voice communication.

Text communication was achieved by simply typing letters or numbers, which were transmitted the moment each was typed, appearing in a speech bubble above the participant's avatar. The text expired after approximately ten seconds from the moment the enter key was pressed. Each participant was provided with a stereo headset for audio communication. The default recording and playback volumes were automatically set using a shell script.

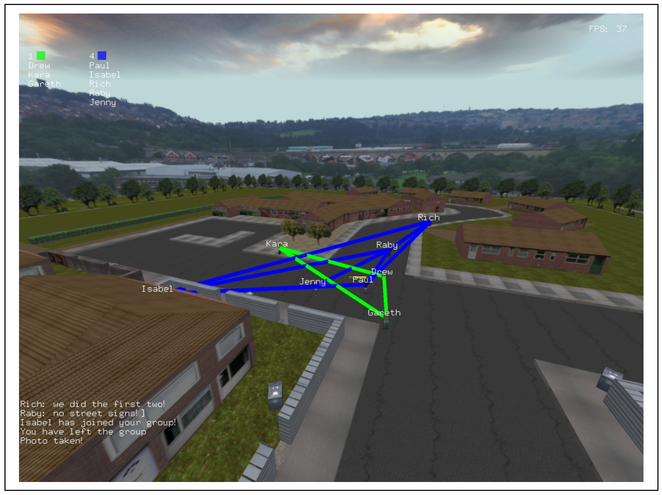


Figure 2: A screenshot from MGDs condition, showing two groups, each linked by different colored lines

MGD functionality used three mouse buttons, and the 'Delete' key to move up one level in the group hierarchy. The display had a crosshair in the middle used for selection. Selecting an avatar with the left mouse button formed/joined a group. Selecting the avatar of a fellow group member with the right mouse button rapidly moved to their location and automatically followed them. Pressing the middle mouse button anywhere moved to the mean location of the group.

Figure 3: A screenshot showing a close-up of the avatars of three participants from the control condition, in front of some terraced houses

4.1.5 Procedure

Several days before the experiment, each participant attended a ten minute preparation meeting, to have photos taken for their avatar, ask questions about the experiment, and read an introductory sheet containing extracts from government urban planning guidelines.

The experiment itself lasted one hour. At the start of the experiment, each participant was provided with three information sheets: another copy of the introductory sheet, instructions that described the CVE's interface, and a schedule for the experiment. They were also provided with an electronic copy of an urban planning report. The report contained the following questions for participants:

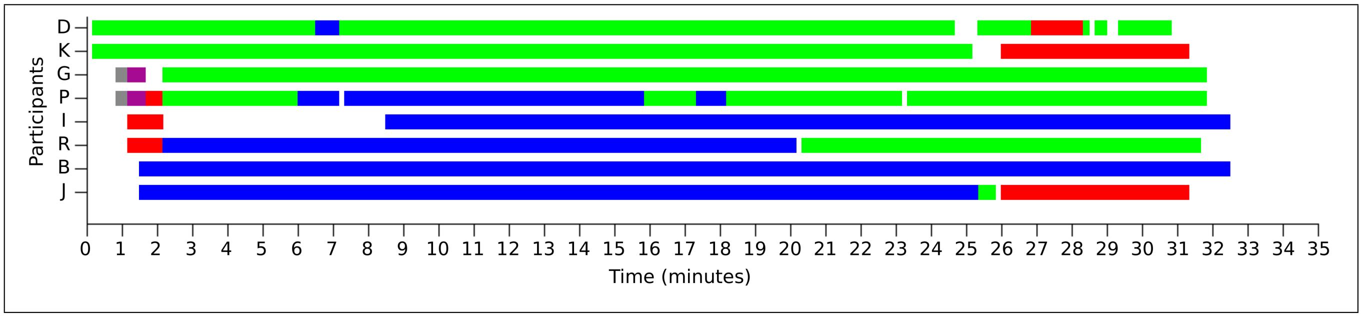


Figure 4: A chart showing which team the MGD participants were in over time. Each team is shown in a different color.

- **Question 1, Permeability:** (a) How many entrance and exit points are there around the estate? What are these for (i.e. cars or pedestrians)? (b) What reduces the speed/volume of traffic? (c) Are there suitable pedestrian routes around the environment? (d) Are the blocks small enough or do you have to walk too far before you reach a choice of direction?

- **Question 2, Character:** (a) Which parts of the environment follow the same pattern/building structure? (b) Find a part of the environment that is not consistent with the layout of the estate. (c) Is this acceptable or should it be changed? (d) Does the estate have character?

- **Question 3, Safety & Security:** (a) Comment on the safety and security of the estate based on your own thoughts, the information in the guidelines and your discussion with other participants. (b) Find examples of where public and private space is clearly distinguished and where it isn't. (c) Discuss which part(s) of the estate you think are least safe. (d) Can you find any blank walls that you think should be overlooked to improve the feeling of safety and help prevent graffiti? (e) Try to suggest some improvements with regard to the safety and security of the estate.

The first 15 minutes of the experiment were used for training. Participants were instructed to experiment with all the controls available to them, with the experimenter and assistant on hand to clarify anything if necessary. Participants logged into a training environment. This contained a 3D representation of a city, of which an area of approximately 75x75m could be explored. There was a main road area, surrounded by large tower-blocks, with small alley ways around the back of them. Two of the tower-blocks could be entered, one from the road, and the other by descending some steps and going under the road in a subway. There was a lift up to the top of one of the blocks.

The next 35 minutes were allocated for the main task – the review of the residential estate. Participants logged into the test environment and traveled around the estate to answer the questions and complete their urban planning report. If a participant came across something relevant to the report, they could take a screenshot of it. The screenshot would simply capture what they were looking at, in the same view that the participant had (i.e. over-the-shoulder, or bird's-eye).

The participants received verbal warnings when there were 10 minutes and 5 minutes remaining on the main task, to encourage them to finish writing up the report. The final 10 minutes were allocated to submitting the report, filling in a questionnaire, and receiving payment.

5 Results

The data collected can be divided into two categories, taskwork and teamwork – 'the work of working together' [1]. The sources of data were the participants' urban planning report sheets, the questionnaires and the server's recording of everything that took place in the environment (text and audio communication, movement, and the makeup of the teams). The report sheets provided data about the taskwork, and the questionnaires and server's recording provided data about the teamwork.

The server's recording was in the form of a log file. It could be played back, either forwards or backwards (rewinding) at various speeds, and with the ability to move the viewpoint around the environment to view the playback at any position or orientation.

Statistical analyzes were performed using independent samples t-tests to compare participants who had been provided with the MGD functionality with those who had not.

5.1 Taskwork

The reports were marked like an exam, according to a mark scheme with example answers.

The mean marks were 16.9 out of 24 ($SD = 5.1$) for the MGDs condition, and 17.3 ($SD = 4.0$) for the control condition. An independent samples t-test showed there was not a significant difference in the taskwork scores of the two groups of participants, $t(16) = 0.20, p > .05$.

The task itself was only of modest difficulty, so it was to be expected that performance would not differ between the two conditions. However, our primary interest lay in how MGDs affected the way in which participants tackled the task.

5.2 Teamwork

The analysis of teamwork consisted of a combination of two methods based on those in [15]. The first method was quantitative, in which the communication and spatial positioning between participants were analyzed, and the results for the MGDs and control conditions were compared. The second method was qualitative, an 'analysis of interaction fragments' [15] (p. 661), in which the paths of the core participants in the MGDs condition were analyzed to draw out patterns of interaction.

5.2.1 Quantitative analysis

For the MGDs condition, each explicit group of participants was given a unique color. This 'team' color remained the same despite changes in the combinations of participants who belonged to that team. The teams are shown in Figure 4. The participants are shown on the y axis, and given a color depending on which team they belong to at each point in time, where time is shown on the x axis. The time of zero represents the time that the server was started. Teams were formed from scratch five times, four times implicitly (pairs of participants walked within 1m of each other) and once explicitly (one participant selected another). The chart shows that for the majority of the experiment there were two teams, one blue and the other green, with participants occasionally switching from one to the other.

The data about participants' movements through the environment were used to calculate how far each participant was from their nearest neighbor every second during the experiment. This was then used to determine the percentage of time participants spent separated by given distances from the other participants (see Figure 5). These data show that participants spent nearly twice as much time within 10m of others when MGDs were provided.

The mean distance to the nearest neighbor was calculated for each participant in both conditions. The overall means were 19.7m for the MGDs condition ($SD = 4.2$) and 25.4m for the control condition ($SD = 3.8$). An independent samples t-test showed that there was a significant difference in the distances to the nearest neighbor for the two conditions, $t(16) = 3.05, p < .01$.

The questionnaire was used to gather data on the use of MGDs. In particular, the automatic following mechanism could be used to rapidly move to a group member's location. Six out of the eight participants said they used the functionality in this way.

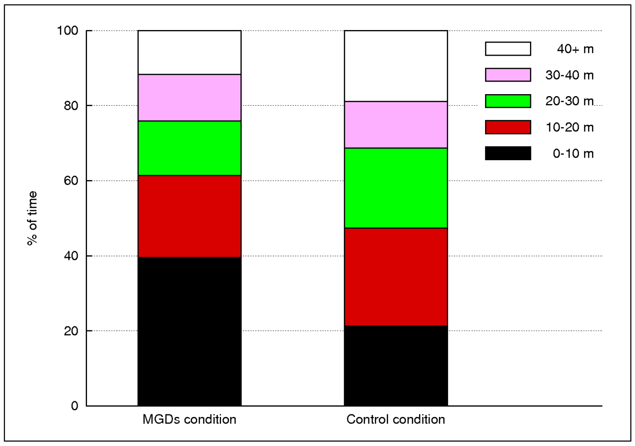


Figure 5: Proportion of time participants spent within given distance of their nearest neighbor

For each batch of participants (the MGDs and control conditions), participant's spoken and text communication was transcripted and analyzed using a communication coding approach [4] to classify each utterance as one of the following:

(a) **Greetings** (e.g. 'Hello *R*!', 'How you doing?')

(b) **Functionality** – communication regarding the system and the groups (e.g. '*D* are you following me?', 'Press home to get a better view', 'Can everyone hear me even though we're in different groups?')

(c) **Environment** – discussion about the 3D world, but not in relation to the task (e.g. 'So is this meant to be an actual part of Leeds?', 'There's Leeds city council bins')

(d) **Task related** (e.g. 'What do you think reduces the speed round here?', 'I've found a bit of the estate that doesn't really match the rest')

(e) **Idle chat** (e.g. '*D* I can actually read what's on your T-shirt!')

Overall there were 133 utterances in the MGDs condition, of which 40 were text-based and 93 were spoken. The utterances occurred in 22 blocks of conversation and in 15 of these, all the speakers were in the same team. There were 18 utterances in the control condition, of which 16 were text-based and 2 were spoken. These utterances occurred in 3 blocks of conversation. These data show that there was much more communication in the MGDs condition, and most of it was task-related (see Figure 6).

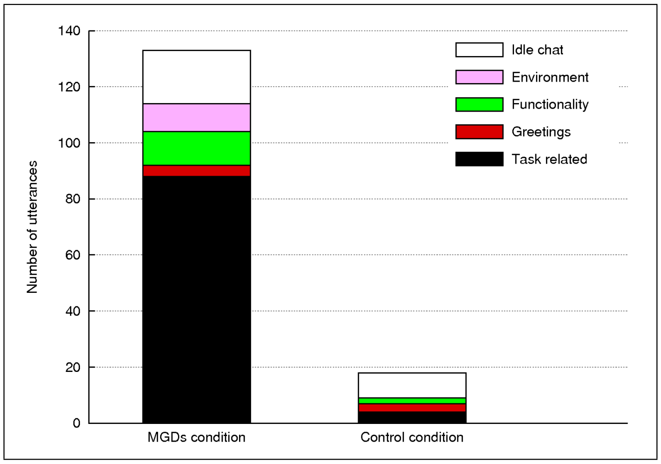


Figure 6: The number of utterances in each communication category for the two conditions

5.2.2 Paths during teamwork

In the MGDs condition, the most persistent combination of team members was *D*, *K* and *G*, in the green team, and *P*, *I*, *R*, *B* and *J*, in the blue team (see Figure 4).

D and *K* were identified as the core members of the green team because they communicated the most. *D* spoke 29 utterances, *K* spoke 22 but *G* (the third member) only spoke 12 utterances.

R and *B* were identified as the core members of the blue team. *R* spoke 41 utterances and *B* spoke 19, which was far greater than the other members *P*, *I* and *J* who spoke 5, 0 and 5 utterances respectively.

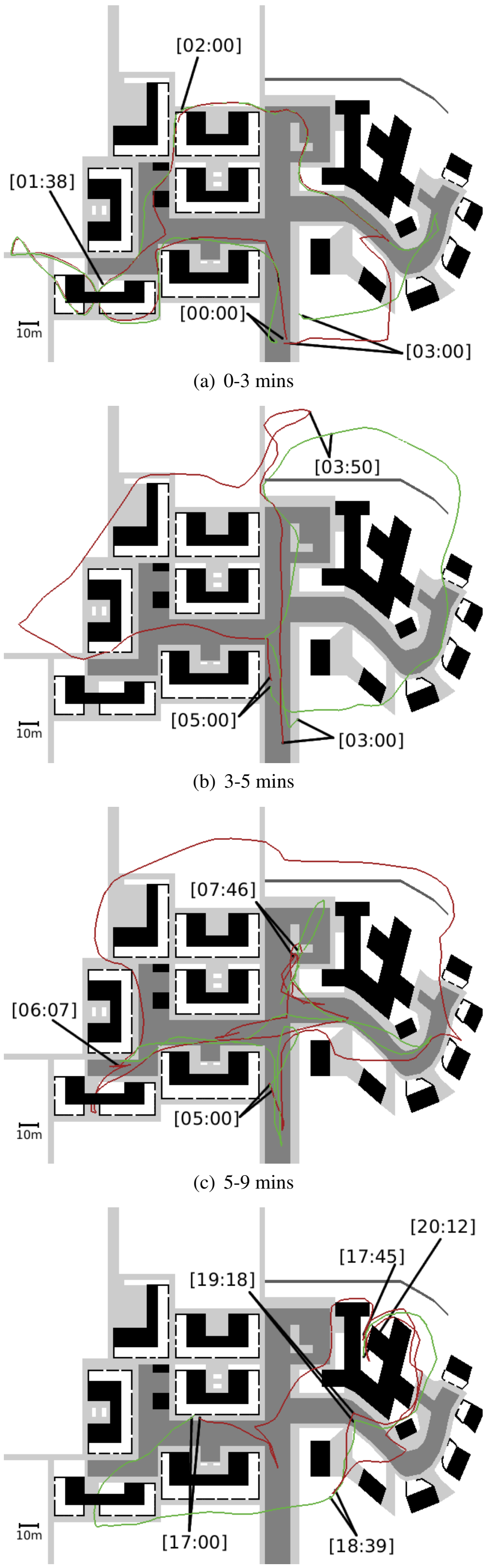
The paths of these core participants from the MGDs condition were analyzed in detail and showed that they sometimes moved together around the environment answering a question, and on other occasions split up to explore their surroundings, and then regrouped to discuss their findings. By contrast, participants in the control condition communicated far less and spent little time in close proximity (see section 5.2.1).

The following paths and communication from the green team illustrate the types of behavior that occurred when MGDs were provided. Figure 7(a): The two core members of the green team started at the entrance to the estate (shown with a timestamp [00:00] in the diagram), navigated around the environment together in a clockwise direction, and returned to the starting point. *D* was following *K* using the automatic following MGD functionality. Their conversation was based on the functionality of the system (the leader/follower mechanism), and the real world location of the virtual environment. The points at which the conversation took place are shown by timestamps on the diagrams and in the extracts below.

```
[01:38] K: D are you following me?
[01:42] D: I am, yes!
[01:45] K: Wicked!
[01:50] R: I think I can see my house
from here!
[02:00] D: So is this meant to be an
actual part of Leeds?
```

Figure 7(b): The core members returned to their starting point [03:00]. They *split up* [03:50] and navigated one side of the estate each, until they regrouped again in the middle [05:00].

Figure 7(c): The core members *split up* again, *D* navigated the perimeter of the environment and *K* stuck to the roads. *K* met the two core participants *R* and *B* from the blue team and joined in their conversation [06:07].



(a) 0-3 mins

(b) 3-5 mins

(c) 5-9 mins

(d) 17-20:12 mins

Figure 7: Paths of the core members K and D from the green team. K and D are represented by green and red lines respectively

[05:47] R: What do you reckon stops the volume of traffic?
[05:57] B: I don't know
[06:03] R: Could it be that it's so windy?
[06:07] K: Dead ends as...
[06:11] R: Was that... was that J?
[06:14] K: No that was K!

The two core members of the green team *regrouped* at time [07:46]. One of the core members of the blue team, *R*, was with them and joined in their conversation. *R* reported the findings from the blue team.

[07:46]/Text D: Hadn't we better start answering some of the questions?
[08:07]/Text K: i already ahve
[08:10]/Text K: haahhha
[08:18]/Text G: probably
[08:36]/Text R: we did the first two!
[08:38]/Text G: how many exits are there?

[08:46] D: I've only found one.
[08:49] R: One what?
[08:52] D: One exit.
[08:54] R: We found four pedestrian and one for cars.
[09:05] R: It's a small world.
[09:08] D: Too true.

The time from [09:08] to [17:00] has been omitted because there was little communication between the core members of the green team throughout this time (two utterances from *D* and one from *K*).

Figure 7(d): The two core members of the green team *split up*, *D* found something of interest [17:45], they *regrouped* [18:39], *D* showed the rest of the team the point of interest from a distance using the bird's-eye view [19:18].

[18:39] D: I've found a bit of the estate that doesn't really match the rest.
[18:42] K: Yeah. So have I.
[18:46] D: What have you put for that?
[18:49] K: One of the two level houses has got a different color wall to the [K is referring to the brown fence around the terraced houses (determined from K's report)]
[18:55] D: Oh, is that it?
[18:57] K: Yeah. Why? What have you got?
[19:00] D: If you press 'Home' and follow me I'll show you.
[19:03] K: OK.

D lead *K* (and *G* who was listening in) to the large building, and stopped by the side of it to talk [19:18]. (Pressing the 'Home' key toggled bird's-eye view).

[19:18] D: If you all look to my left now, and have a look with 'Home'...
[19:23] K: OK.
[19:24] D: It's laid out a completely different way and there's a dead end in the middle.
[19:33]/Text G: ah yeah i see
[19:34] K: Oh yeah!

K and *G* then followed *D* to the point of interest.

[20:12] K: I see what you mean.

6 Discussion

Our goal was to develop techniques for Mobile Group Dynamics that helped people work together over an extended period of time, in a large-scale space. MGDs had a neutral effect on task performance (the task was achievable by oneself) but did produce fundamental changes in the way participants went about performing the task and the quantity of teamwork that took place. In particular, this was shown by the amount of time that participants spent near each other, the way they continued to collaborate after periods of separation, and the amount of communication that took place.

Participants in the MGDs condition spent much more time in close proximity (within 10m of their nearest neighbor for 40% of the experiment) than participants in the control condition (21%), and two aspects of MGDs contributed to this. Firstly, participants could easily identify fellow group members because lines between group members indicated the location of others and each group was given a unique color (see soccer team analogy in Section 2.1). Secondly, the automatic following functionality helped people remain together while they traveled, and also provided an easy way of regrouping with one's fellow members (75% of the MGDs participants used the functionality in this way).

It is suggested that 'cognitive ease' as well as functionality affects group behavior in CVEs [2], and this may explain why MGDs were so successful at helping participants collaborate over an extended time that included periods of separation. Firstly, allowing groups to form automatically via spatial proximity minimized the effort involved of initially forming a collaboration with other participants (80% of groups were formed in this way). Secondly, the explicit indication of who was in each team (see above) and the fact that audio communication within a group was not attenuated by distance meant that participants did not lose contact if they wandered away from their fellow group members. Thirdly, leaving or switching groups had to be done explicitly and, therefore, was effortful.

There were over seven times the number of utterances in the MGDs condition, compared to the control condition. This is the result of the suite of techniques as a whole. It could be argued that the very presence of MGDs would have given participants an idea of how to work together effectively [13] and, with 66% of the conversation being task related, this was representative of the extra teamwork taking place.

Finally, although participants could communicate with group members wherever they were in the environment, they still preferred to spatially *regroup* to discuss their findings. When there was a point of interest, it seemed important for everyone to see it from the same viewpoint and get a shared understanding of it (see the dialog in Section 5.2.2, for Figure 7(d)). We plan to address this by further research into techniques to improve awareness of who can hear you and who is talking, allow rapid movement to another location by teleporting, and provide multiple views so participants can see what their group members are looking at.

Acknowledgements

The first author is funded by a Doctoral Training Grant Studentship from the School of Computing, University of Leeds.

References

[1] K. Baker, S. Greenberg, and C. Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work*, pages 96–105, New Orleans, Louisiana, USA, November 16-20 2002. ACM.

[2] B. Becker and G. Mark. Social conventions in computer-mediated communication: A comparison of three online shared virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 19–39. Springer, London, 2002.

[3] J. Blascovich. A theoretical model of social influence for increasing the utility of collaborative virtual environments. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 25–30, Bonn, Germany, September 30-October 2 2002. ACM.

[4] C. A. Bowers, F. Jentsch, E. Salas, and C. C. Braun. Analyzing communication sequences for team training needs assessment. *Human Factors*, 40(4):672–679, 1998.

[5] B. Brown and M. Bell. CSCW at play: 'There' as a collaborative virtual environment. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*, pages 350–359, Chicago, Illinois, USA, November 2004. ACM.

[6] L. Cheng, S. Farnham, and L. Stone. Lessons learned: Building and deploying shared virtual environments. In R. Schroeder, editor, *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, pages 90–111. Springer, London, 2002.

[7] L. Chittaro and S. Burigat. 3D location pointing as a navigational aid for virtual environments. In *Proceedings of the 7th International Conference on Advanced Visual Interfaces (AVI '04)*, pages 267–274, Gallipoli, Italy, May 2004. ACM.

[8] Commission for Architecture and the Built Environment. The cost of bad design. `http://www.cabe.org.uk/AssetLibrary/8125.pdf`, 2006. (Accessed 27 April 2007).

[9] C. Greenhalgh and S. Benford. Supporting rich and dynamic communication in large-scale collaborative virtual environments. *Presence: Teleoperators and Virtual Environments*, 8(1):14–35, 1999.

[10] G. Hiebert. Openal 1.1 specification and reference. `http://www.openal.org/openal_webstf/specs/OpenAL11Specification.pdf`, 2005. (Accessed 17 July 2007).

[11] J. Hindmarsh, M. Fraser, C. Heath, S. Benford, and C. Greenhalgh. Object-focused interaction in collaborative virtual environments. *ACM Transactions on Computer-Human Interaction*, 7(4):477–509, 2000.

[12] K. Lewin. *Resolving social conflicts : selected papers on group dynamics*. Research Center for Group Dynamics, University of Michigan. Harper & Brothers, New York, 1948. Gertrud Weiss Lewin, Ed.

[13] J. M. Linebarger, C. D. Janneck, and G. D. Kessler. Leaving the world behind: Supporting group collaboration patterns in a shared virtual environment for product design. *Presence: Teleoperators and Virtual Environments*, 14(6):697–719, 2005.

[14] Office of the Deputy Prime Minister. Urban design in the planning system: Towards better practice. `http://www.odpm.gov.uk/index.asp?id=1145239`, 2000. (Accessed 8 February 2006).

[15] R. Schroeder, I. Heldal, and J. Tromp. The usability of collaborative virtual environments and methods for the analysis of interaction. *Presence: Teleoperators and Virtual Environments*, 15(6):655–667, 2006.

[16] S. Singhal and M. Zyda. *Networked virtual environments : design and implementation*. Addison-Wesley, Reading, MA ; Harlow, England, 1999.

[17] A. Steed, M. Slater, A. Sadagic, A. Bullock, and J. Tromp. Leadership and collaboration in shared virtual environments. In *Proceedings of the IEEE Virtual Reality (VR '99)*, page 112, Houston, TX, USA, March 13-17 1999. IEEE.

[18] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics*, 23(3):249–258, 2004.

[19] B. W. Tuckman. Developmental sequence in small groups. *Psychological Bulletin*, 63(6):384–399, 1965.

[20] D. L. Weatherford. Representing and manipulating spatial information from different environments: Models to neighborhoods. In R. Cohen, editor, *The Development of Spatial Cognition*, pages 41–70. Lawrence Erlbaum Associates, Florence, Kentucky, 1985.

[21] H. Yang and G. M. Olson. Exploring collaborative navigation: the effect of perspectives on group performance. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)*, pages 135–142, Bonn, Germany, September 30-October 2 2002. ACM.

EGVE Symposium (2008)
B. Mohler and R. van Liere (Editors)

Using Teleporting, Awareness and Multiple Views to Improve Teamwork in Collaborative Virtual Environments

Trevor J. Dodds† and Roy A. Ruddle‡

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

Abstract

Mobile Group Dynamics (MGDs) are a suite of techniques that help people work together in large-scale collaborative virtual environments (CVEs). The present paper describes the implementation and evaluation of three additional MGDs techniques (teleporting, awareness and multiple views) which, when combined, produced a 4 times increase in the amount that participants communicated in a CVE and also significantly increased the extent to which participants communicated over extended distances in the CVE. The MGDs were evaluated using an urban planning scenario using groups of either seven (teleporting + awareness) or eight (teleporting + awareness + multiple views) participants. The study has implications for CVE designers, because it provides quantitative and qualitative data about how teleporting, awareness and multiple views improve groupwork in CVEs.

Categories and Subject Descriptors (according to ACM CCS): C.2.4 [Computer-Communication Networks]: Distributed Systems – Distributed applications; H.1.2 [Models and Principles]: User/Machine Systems – Human factors; Software psychology; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented and virtual realities; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – Collaborative computing; Computer-supported cooperative work; Synchronous interaction; I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism – Virtual Reality

1. Introduction

Previous research developed techniques called Mobile Group Dynamics (MGDs), which helped groups of people work together while they traveled around large-scale collaborative virtual environments (CVEs) [DR08]. Compared to a conventional CVE, these techniques led to a seven-fold increase in the amount of communication that took place between participants. 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 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.

The present paper describes how these shortcomings were tackled by adding new functionality to MGDs. We were reminded that 'CVEs...do not necessarily need to reflect or embody the characteristics of conventional environments to enable them to support particular forms of activity or interaction' [FGV*00], and 'This is the trap VR often falls into; VR tries to imitate reality...' [Pek02]. Therefore, the new functionality took advantage of the fact that CVEs do not need to be limited to real world constraints. Three types of functionality were implemented.

First, our hypothesis was that participants collocated to communicate with group members because, given that the CVE used directional sound, participants assumed that the audio was also distance attenuated. In fact, distance attenuation was disabled for intra-group communication, as was explicitly stated in participants' verbal and written instruc-

† e-mail: trev@comp.leeds.ac.uk
‡ e-mail: royr@comp.leeds.ac.uk

tions. To overcome this, we developed 'awareness' functionality that provided visual feedback about who was receiving audio at a given moment in time, and who was speaking.

Second, it is well known that in CVEs participants often find it difficult to understand what each other is looking at [HFH*00]. To overcome this, a participant's own (main) view was supplemented by small viewports that showed the views of fellow group members.

Third, 'walking' is time consuming, so teleporting functionality was added. This allowed participants to move directly to any point in the environment by clicking on the appropriate location on the display, or move directly to another participant's position by clicking on 'their' viewport.

The following sections describe further background research, the implementation of the new MGDs functionality, and its evaluation using an urban planning scenario.

2. Background

This section briefly reviews research relevant to teleporting, promoting awareness of others and the use of multiple views in virtual environments. People often wish to spend large amounts of time exploring an environment, be it virtual or real. To minimize the amount of time spent traveling in the real world people may choose to drive rather than wait for a bus, or run instead of walk. In electronic environments, however, we can make use of non-naturalistic forms of movement. For example, using a search engine we can generate hyperlinks that jump straight to task-related pages.

An evaluation of the use of hyperlinks in virtual environments showed that there was a speed-accuracy tradeoff when compared to conventional navigation (walking) [RHPJ00]. Put simply, participants found their target locations faster with hyperlinks, but visited more locations in the process. This suggests that the efficiency of teleporting is in the speed, so CVE designers might infer from this that *instantaneous* teleportation should be implemented to get the most out of the speed increase. However, [RHPJ00] also highlights the importance visual continuity. This is something that instant teleportation can lose, e.g. teleporting between self-contained areas in chat-based VEs such as There[†] and Second Life[‡].

Elvins et al. [ENSK01] helped overcome visual discontinuity by providing 'worldlets', or small 3D thumbnails, of landmarks which the user could teleport to. This was more useful to users than a 2D image or textual description of the destinations. These worldlets overcame the visual discontinuity that can occur when a user teleports from one place to another, but other techniques are needed to provide information about the spatial relationship of the two places and the rapid controlled movement used to implement teleporting in the present study is one method of achieving this.

Participants in desktop VEs 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 [HFH*00]. 2) What you see is *not* what I see, which makes it difficult to understand another's perspective.

Problem 1 is likely to happen because of a narrow field of view. This is inherent in desktop VEs wanting to minimize distortion to keep realism [FGV*00]. Problem 2 is due to a removal of real world sensory data, such as eye movements and depth perception. This problem tends to be compensated for by a large increase in the amount of verbal communication that takes place [RSJ02].

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 [YO02], 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. [WSWL02] 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 didn't interfere with the other participant. Sonnenwald et al. [SWM03] 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, our hypothesis was that by providing multiple viewports to the user, MGDs would provide the best of both worlds.

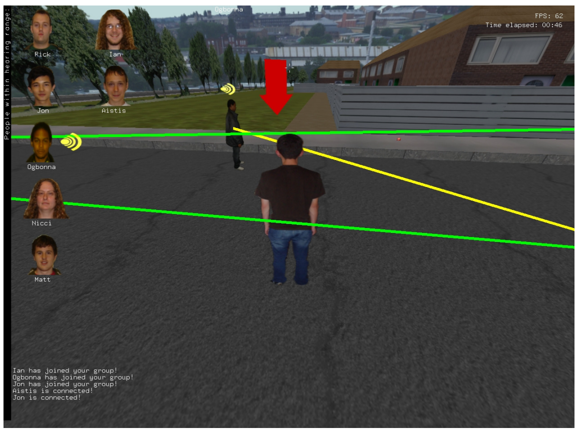
Real time views of other users' perspectives have been used many times in multiplayer games, such as console games that are designed to be played together on a single display. In this case, however, the views are aimed at different people (e.g. looking at another person's view could be considered cheating). The 'split screen' is used to provide a cheap alternative to the players, instead of requiring that they have multiple visual display units. The present study investigated the provision of multiple views to each user, via a main window and thumbnail views of other participants. Participants were allowed to click on these thumbnails to teleport to the appropriate person.

† http://www.there.com (Accessed 23/1/2008)

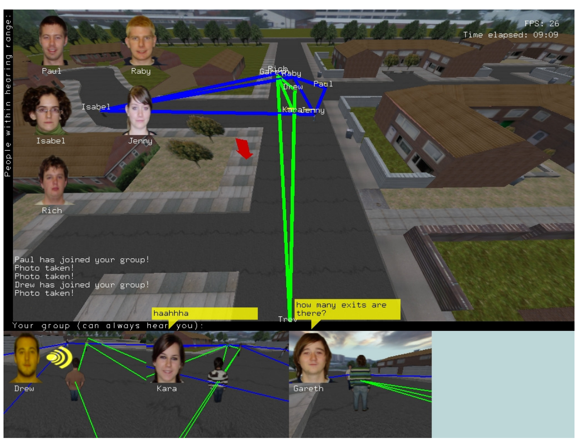
‡ http://www.secondlife.com (Accessed 23/1/2008)

3. Implementing Mobile Group Dynamics

The basic MGDs techniques incorporated an explicit hierarchical grouping system, represented using a 'group graph' metaphor, and methods to assist movement as a group. For this study, we added awareness of who was talking and who would receive audio (i.e. participants within hearing range; this included all participants in one's own group because there was no distance attenuation for intra-group audio communication), teleporting functionality and multiple views (including the ability to teleport to a fellow group member by clicking on their view). This new functionality, along with the group graph, can be seen in Figure 1.



(a) Teleport condition, shown in the over-the-shoulder perspective



(b) Multiple views condition, shown in the bird's-eye view

Figure 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 in (b)*

The faces of all participants who were within hearing range were displayed on the Head-Up Display (HUD). These faces were photographs of the participants (extracted from their photographic avatars), so they could be easily recognized. When a participant spoke, the faces were highlighted. 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.

The teleporting functionality took place as rapid controlled movement, to help prevent disorientation associated with an instantaneous change of location (see Section 2). It utilized the same algorithm as the automatic following functionality from the original MGDs which could also be used to rapidly move to another's location (even if the other was a moving target). Inspiration for the algorithm was taken from [MCR90], with the addition of gradual acceleration as well as deceleration. To avoid breaks in visual continuity caused by teleporting through walls, our implementation raised a participant to a birds-eye view during teleportation, so the participant could clearly see where they were being taken.

4. Experiment

The experiment used the context of urban planning, with participants asked to use a CVE to review the design of a new housing estate. Participants were run in two batches. In the first of these (the teleporting condition), participants had all the basic MGD functionality from [DR08], and new MGD functionality to provide awareness of who was talking, who was within hearing range and teleporting. Participants in the second batch were provided with multiple views (the multiple views condition), in addition to all the MGD functionality that was provided to the other batch of participants.

4.1. Method

The experiment took place in an undergraduate computing laboratory. Each participant was provided with a headset, and they were spread out across the laboratory so they could only communicate using audio and text communication from within the environment. Participants used two adjacent computers, one for the CVE and the other for the urban planning report write-up. The CVE application, environment and experimental procedure were the same as in [DR08].

4.1.1. Participants

All participants were undergraduate students from the School of Computing, who had not taken part in the previous study. Eight participants were recruited for each run, but one participant in the teleporting condition was unavailable on the day of the experiment. The remaining seven participants in the teleporting condition (6 men and 1 woman) had a mean age of 21.7 ($SD = 5.2$). The eight participants in the multiple views condition (5 men and 3 women) had a mean age of 21.8 ($SD = 4.1$).

All the participants volunteered for the experiment, gave informed consent and were paid an honorarium for their participation.

4.1.2. CVE application

The software application and 3D sound model are described in the previous study [DR08].

Distance attenuation was turned off for communications between members of the same group. This was clarified by displaying photographs of the faces of participants who would receive any transmitted audio. These faces were displayed on the HUD, and were added and removed appropriately as participants changed their position in the environment and switched groups. In addition, an icon was placed above a participant's avatar, and by the side of their face on the HUD, when they were talking.

4.1.3. Environment

The environment was a residential estate that was based on a real estate in Leeds. An annotated map of the estate is shown in Figure 2.

All participants were represented in the environment with a photographic avatar (using four photos: front, back, left and right). Participants were given an over-the-shoulder perspective, with the option of switching to and from a bird's-eye view. An over-the-shoulder perspective meant that participants could see each other relative to their avatar, and be more aware of how others perceived them [CFS02].

4.1.4. User Interface

The participants used desktop workstations, and a two-handed control method, with one hand on the keyboard and the other hand on a 3-button mouse. By holding down appropriate arrow keys a participant could move forward/backward/left/right at 6 m/s, and heading and pitch could be changed by moving the mouse. This is a common gaming control method (e.g. [BB04]).

The 'Insert' key was used to take screenshots, the 'Home' key to toggle between over-the-shoulder and bird's-eye views, and holding down the 'Page Down' key allowed the participant to use voice communication.

Text communication was achieved by simply typing letters or numbers, which were transmitted the moment each was typed, appearing in a speech bubble above the participant's avatar. The text expired after approximately ten seconds from the moment the enter key was pressed. Each participant was provided with a stereo headset for audio communication. The default recording and playback volumes were automatically set using a shell script.

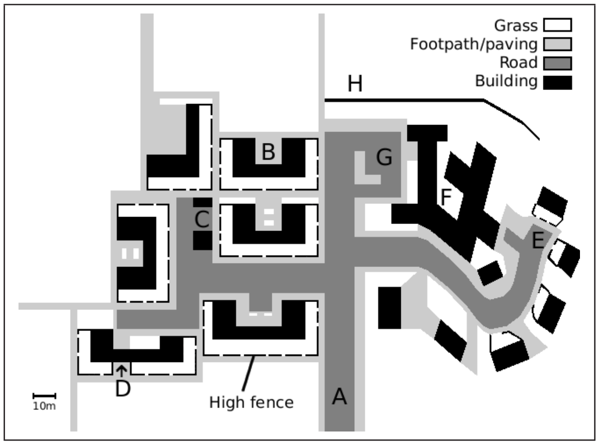


Figure 2: *A map of the estate. The estate had an entrance road in the middle (point A), which acted as a dividing line between two styles of building. On the left-hand side of the entrance road, there were brown-bricked terraced houses, which were mostly horse-shoe shapes creating partially enclosed private space (e.g. point B). The front gardens were bordered by high fences, and there were six garages in the road (C). There was an archway under one of the terraces (D). On the right-hand side of the entrance road there were red-bricked bungalows (single story buildings) along the edge of the curved road, with gardens bordered by low brick walls (e.g. E). There was a single-story care home for elderly people (F), with a car park to the left with space for six cars (G), and a hedge-row above it partly separating private land around the care home from public parkland (H).*

The basic MGD functionality used three mouse buttons, and the 'Delete' key to move up one level in the group hierarchy. The display had a crosshair in the middle used for selection. Selecting an avatar with the left mouse button formed/joined a group. Selecting the avatar of a fellow group member with the right mouse button rapidly moved to their location and automatically followed them. Pressing the middle mouse button anywhere moved to the mean location of the group.

Holding down the numpad zero key released the mouse from controlling heading and pitch, and allowed it to control the position of the red teleporting arrow. Once the arrow was positioned in the desired location, a left mouse click teleported the participant there.

Participants in the multiple views condition could position the teleporting arrow over one of their group members views, and clicking the left mouse button would teleport them to that group member's location. By default, the participant's subsequent movements were tethered to that group member (the automatic following functionality in basic MGDs) but the participant could 'free' themselves simply by pressing a movement key.

The multiple views took up the bottom quarter of the screen. A limit was imposed of three views, each taking up a quarter of the horizontal space, with the remaining quarter reserved for displaying the faces of any other group members. These could be selected using the numpad zero key to release the mouse pointer. Selecting them showed their view in one of the existing viewports, swapping out the member whos view had been replaced.

4.1.5. Procedure

A 10 minute meeting was held with participants a few days before the experiment. They received a verbal explanation of the experiment, a single-sided A4 sheet containing extracts from UK urban planning guidelines and a consent form. They also had photos taken for their avatar during this time.

The experiment itself lasted one hour. At the start participants were provided with another copy of the urban planning guidelines sheet, an instruction sheet for using the CVE, an experiment schedule, and an electronic copy of an urban planning report which they had to complete during the experiment. The report contained the following questions, which participants were asked to illustrate using screenshots:

- **Question 1, Permeability:** (a) How many entrance and exit points are there around the estate? What are these for (i.e. cars or pedestrians)? (b) What reduces the speed/volume of traffic? (c) Are there suitable pedestrian routes around the environment? (d) Are the blocks small enough or do you have to walk too far before you reach a choice of direction?
- **Question 2, Character:** (a) Which parts of the environment follow the same pattern/building structure? (b) Find a part of the environment that is not consistent with the layout of the estate. (c) Is this acceptable or should it be changed? (d) Does the estate have character?
- **Question 3, Safety & Security:** (a) Comment on the safety and security of the estate based on your own thoughts, the information in the guidelines and your discussion with other participants. (b) Find examples of where public and private space is clearly distinguished and where it isn't. (c) Discuss which part(s) of the estate you think are least safe. (d) Can you find any blank walls that you think should be overlooked to improve the feeling of safety and help prevent graffiti? (e) Try to suggest some improvements with regard to the safety and security of the estate.

5. Results

There were two types of work that took place in the experiment: taskwork and teamwork [BGG02]. Taskwork refers to the answers given in participants' reports, whereas data about teamwork were provided by the server's log of the movements, communication and groups that participants formed.

The urban planning reports were marked like an exam. Participants names were on the reports, marking wasn't blind. An independent samples t-test showed no significant difference between the teleporting and multiple views conditions, $t(13) = 1.49, p = .16$. Participants in the teleport condition had a mean mark of 18.7 ($SD = 3.3$) out of 24, and 16.3 ($SD = 3.1$) in the multiple views condition. Our focus, however, was on how participants went about doing the task (i.e. the teamwork), and how different MGD functionality affected participants' behavior. This was analyzed both quantitatively and qualitatively.

5.1. Quantitative Analysis

For each batch of participants, the spoken and text communication was transcripted and analyzed using a communication coding approach [BJSB98] to classify each utterance as one of the following:

(a) **Greetings** (e.g. 'Hey *M*!', 'Hi *G*!')
(b) **Functionality** – communication regarding the system and the groups (e.g. 'Think we need smaller groups than all of us!', 'You do realize that if you just press 'Home' you get a bird's-eye view and it's a lot easier to see!')
(c) **Environment** – discussion about the 3D world, but not in relation to the task (e.g. 'I swear you should be able to see uni from here.', 'I kind of might have figured out where the pictures were taken of, you know the Leeds skyscrapers ones.')
(d) **Task related** (e.g. 'Which part's the least safe?', 'I'd say where we're stood now, *J*.')
(e) **Idle chat** (e.g. 'Party at my flat. Come on, let's go!')

These data were analyzed in terms of the quantity of communication that took place, and where participants were relative to each other when they communicated. For comparison, data are provided from a previous study [DR08] when other participants had performed the same urban planning task either in a conventional CVE ('control' in Figure 3) or with basic MGDs functionality (see Figures 3 and 4). Note that the average group size in the basic MGDs, teleport and multiple views conditions was 3.5, 2.5 and 3.0 respectively.

The total number of utterances made by participants in the basic MGDs (data from [DR08]), teleport and multiple views conditions (data from the present study) was analyzed using a univariate analysis of variance (ANOVA). This showed that there was a significant difference between the conditions, $F(2,20) = 3.91, p = .04$. Tukey HSD posthoc tests showed that the difference between basic MGDs and multiple views was significant ($p = .03$) but the other pairwise comparisons were not. The mean amount of communication increased by 226% from the basic MGDs to the teleport condition, and by another 27% from the teleport to the multiple views condition. Within this, task related communication increased by a factor of two from basic MGDs to the teleport and multiple views conditions, but this was not

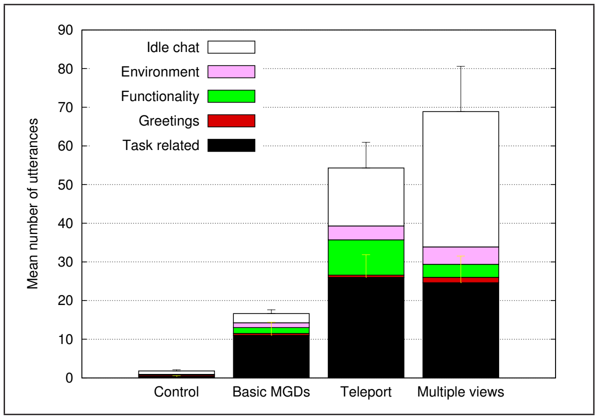


Figure 3: *Mean number of utterances made by the participants in each condition. The control and basic MGDs conditions are from [DR08]. The error bars are shown for task related utterances and idle chat.*

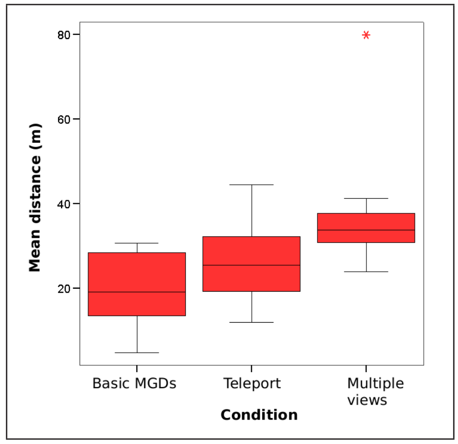


Figure 4: *Mean distance to the nearest group member at the time of each participant's utterances. The basic MGDs condition was from [DR08].*

significant. Idle chat more than doubled from the teleport to the multiple views condition (see Figure 3).

One of the limitations identified in our previous research was that participants tended to assemble in one place in the CVE before communicating, even though this was unnecessary with the basic MGDs functionality that was provided (see Introduction). To determine whether the new functionality provided in the present study overcame this limitation, each time a participant made an utterance the distance to their nearest group member was calculated, and the mean for each participant in the basic MGDs, teleport and multiple views conditions was analyzed using a univariate ANOVA. The two participants who didn't speak at all during the experiment were excluded from the analysis, one was from the basic MGDs condition and the other was from the multiple views condition. The ANOVA showed that there was a significant difference between the conditions, $F(2,18) = 3.56, p = .05$. Tukey HSD posthoc tests showed that the difference between basic MGDs and multiple views was significant ($p = .04$) but the other pairwise comparisons were not (see Figure 4).

5.2. Qualitative Analysis

The quantitative analyses show that teleporting and multiple views increased both the quantity of communication that took place and the distance over which participants communicated. The purpose of the qualitative analysis was to understand the underlying behavioral changes that cause these increases, and how teleporting was used in general.

The server log allowed the distances participants traveled while teleporting and walking to be calculated and showed that, overall, 16% of travel was by teleporting. Further investigation showed that there were two distinct uses of teleporting. First, teleporting was used to speed up exploration of the environment, particularly when participants first entered the environment (see Figure 5). Second, teleporting was used to reach points of interest. For example, at one point during the experiment the some participants' conversation was about blank walls, which was relevant to one of the questions in the task. The blank walls were at the ends of the horseshoe-shaped buildings. Participant *O*, represented by a green line in Figure 6, teleported across the building on the left to view the blank walls (timestamp [26:20]). *O* then teleported up to the top of the map to see the walls that *I* and *R* were talking about.

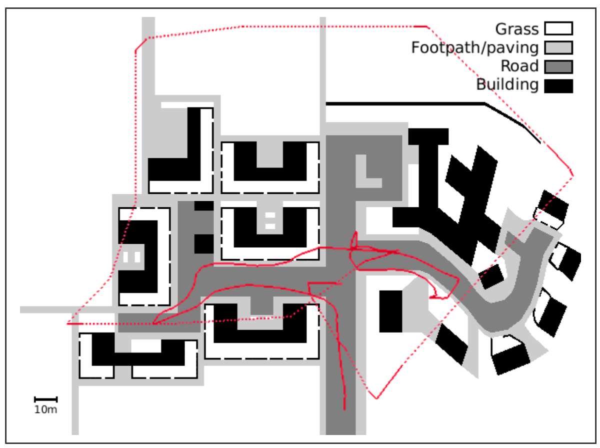


Figure 5: *Path showing the first 5 minutes of movement of a participant who used teleporting to speed up their exploration of the environment. A solid line represents walking, and a dotted line represents teleportation.*

In order to teleport to a point of interest, a participant must first know its location within the environment. This is sometimes difficult, as the following conversation extract from the teleport condition shows:

```
[09:25] O: Look, show me, M, show me the two
entry points then, the road ones at least.
[09:29] M: Alright well, are you where I am now?
[09:35] O: Where are you?
```

The multiple views condition helps with this by allowing participants to teleport to the location of a group member by clicking on their viewport:

```
[39:42] C: That's useful!
[39:43] S: Yeah! Where are you? I'll show you!
[39:49] C: I'll teleport to you! Hang on!
```

Each component of the new MGDs functionality (awareness of who could hear one's communication, multiple views and teleporting) that was provided in the present study had the potential to increase the distance over which participants communicated. The data indicate that multiple views made the greatest contribution (see Figure 4). To identify whether awareness or teleporting was the most important secondary cause a detailed analysis was made of the communication and movement of the two participants (*I* and *O*) who spoke the most in the most persistent group in the teleport condition.

I and *O* both spoke in 18 conversation blocks, but used teleporting in only four of these blocks. On all four of these occasions, *I* and *O* used teleporting to collocate within the environment. In the other blocks *I* and *O* either were together (5 blocks), remained separated (2 blocks), separated without teleporting (3 blocks) or collocated without teleporting (4 blocks). This suggests that the awareness functionality was more important than teleporting for increasing the distance over which participants communicated.

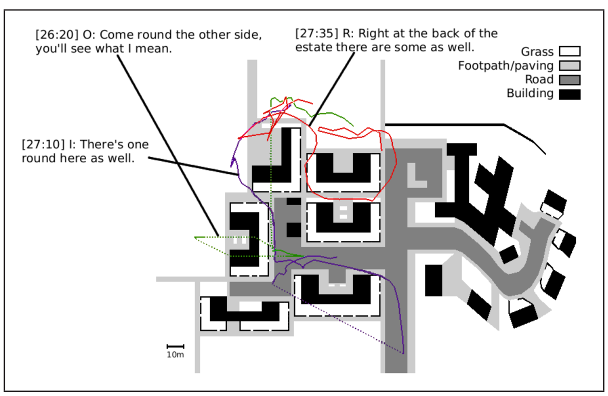


Figure 6: *Paths taken by participants O (green line), I (purple line) and R (red line) when talking about blank walls (a point of interest). The solid lines representing walking, and dotted lines represent teleporting (participant O teleports to point of interests at the ends of the horseshoe-shaped buildings). To place participants' movement in context, the paths are labeled with timestamps and conversation utterances.*

6. Discussion

We identified problems in conventional CVEs of finding other participants in relation to points of interest and understanding their perspective (see problems 1 & 2 in Section 2). The group graph metaphor could help with finding others, since the graph 'tracked' participants and the nodes corresponded to avatars, with edges denoting group membership (see Figure 1). One could find a group member by following a line from their avatar until they reached a node. However, the teleporting and multiple views functionality took this a step further. It allowed participants to teleport directly to a group member of their choice by selecting the appropriate viewport, and the qualitative data gave an example of how this helped participants (see the conversation extract in Section 5.2). Furthermore, providing participants with multiple views specifically tackled the problem of understanding another's perspective.

The qualitative data showed that teleporting was used in two ways: increasing the speed of movement (in particular, an initial speed search) and movement to points of interest. This is a simple method of time-saving functionality, however potential drawbacks must not be overlooked. Firstly, as with any new functionality, a potential problem might be making the system over complicated. Other features may be forgotten about and may not be used to their full potential. Secondly, and perhaps more subtly, teleporting could mean people lose the feel for distance. In an urban planning context, it is important that participants in the CVE get a feel for the scale of the environment, in particular the size of buildings and proposed developments. One of the questions for the urban planning report was asking participants if they thought the blocks of houses were the right size. Financial savings are made by building the houses joined together in blocks, but a large block size decreases permeability of the estate, making it bad for transport and pedestrians (they have to go further before they can change direction). Teleportation may mean participants lose a sense of scale and large blocks could go unnoticed.

One of the places where the original MGDs techniques fell short of their goals was in facilitating communication when participants were spatially separated within the environment. The fact that participants tended to collocate to communicate in the basic MGDs condition was a sign of inefficient groupwork – participants were either taking time to collocate when they wanted to communicate, or they were waiting until they were coincidentally collocated before they said anything. Providing functionality to communicate with group members from a distance, and informing the participants of this in the instructions, was not enough to make their behavior more efficient. This study indicates that by providing feedback to the participants, they became more aware of how the system works. The quantitative data showed that in both the teleporting and multiple views conditions, participants communicated across greater distances than in the basic MGDs condition (see Section 5.1). The interesting thing about this feedback from the system is it's not specifically new functionality in the sense of a new tool at the users' disposal, like teleporting and multiple views are. Instead it provides awareness of *existing* functionality: the ability to communicate with group members from a distance. As Schroeder et al. reflect, do we improve usability 'by means of improving the systems and features of the environment, or by improving the users' awareness of their activities and settings?' [SHT06] (p. 666).

Finally, in previous research, participants communicated a great deal to overcome the lack of sensory information that CVEs provided [HFH*00] [RSJ02]. However, in the present study participants communicated much more than in conventional CVEs because they were provided with more sensory information (e.g. awareness of who could hear you and who was speaking, and multiple views providing an 'extra pair of eyes'). The quantitative data showed that the amount of conversation increased 4 times from the basic MGDs condition to the multiple views condition. This increase in communication was indicative of more teamwork taking place.

Acknowledgments

The first author is funded by a Doctoral Training Grant Studentship from the School of Computing, University of Leeds.

References

[BB04] Brown B., Bell M.: CSCW at play: 'There' as a collaborative virtual environment. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)* (Chicago, Illinois, USA, November 2004), ACM, pp. 350–359.

[BGG02] Baker K., Greenberg S., Gutwin C.: Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work (CSCW '02)* (New Orleans, Louisiana, USA, November 16-20 2002), ACM, pp. 96–105.

[BJSB98] Bowers C. A., Jentsch F., Salas E., Braun C. C.: Analyzing communication sequences for team training needs assessment. *Human Factors 40*, 4 (1998), 672–679.

[CFS02] Cheng L., Farnham S., Stone L.: Lessons learned: Building and deploying shared virtual environments. In *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*, Schroeder R., (Ed.). Springer, London, 2002, pp. 90–111.

[DR08] Dodds T. J., Ruddle R. A.: Mobile group dynamics in large-scale collaborative virtual environments. In *Proceedings of the IEEE Virtual Reality (VR '08)* (Reno, Nevada, USA, March 8-12 2008). In press.

[ENSK01] Elvins T. T., Nadeau D. R., Schul R., Kirsh D.: Worldlets: 3-D thumbnails for wayfinding in large virtual worlds. *Presence: Teleoperators and Virtual Environments 10*, 6 (2001), 565–582.

[FGV*00] Fraser M., Glover T., Vaghi I., Benford S., Greenhalgh C.: Revealing the realities of collaborative virtual reality. In *Proceedings of the 3rd International Conference on Collaborative Virtual Environments (CVE '00)* (San Francisco, California, September 2000), ACM, pp. 29–37.

[HFH*00] Hindmarsh J., Fraser M., Heath C., Benford S., Greenhalgh C.: Object-focused interaction in collaborative virtual environments. *ACM Transactions on Computer-Human Interaction 7*, 4 (2000), 477–509.

[MCR90] Mackinlay J. D., Card S. K., Robertson G. G.: Rapid controlled movement through a virtual 3D workspace. *Computer Graphics 24*, 4 (1990), 171–176.

[Pek02] Pekkola S.: Critical approach to 3D virtual realities for group work. *Proceedings of the Second Nordic Conference on Human-Computer Interaction (NordCHI '02)* (October 2002), 129–138.

[RHPJ00] Ruddle R. A., Howes A., Payne S. J., Jones D. M.: The effects of hyperlinks on navigation in virtual environments. *International Journal of Human-Computer Studies 53*, 4 (2000), 551–581.

[RSJ02] Ruddle R. A., Savage J. C. D., Jones D. M.: Verbal communication during cooperative object manipulation. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)* (Bonn, Germany, September 30-October 2 2002), ACM, pp. 120–127.

[SHT06] Schroeder R., Heldal I., Tromp J.: The usability of collaborative virtual environments and methods for the analysis of interaction. *Presence: Teleoperators and Virtual Environments 15*, 6 (2006), 655–667.

[SWM03] Sonnenwald D. H., Whitton M. C., Maglaughlin K. L.: Evaluating a scientific collaboratory: Results of a controlled experiment. *ACM Transactions on Computer-Human Interaction 10*, 2 (2003), 150–176.

[WSWL02] Wössner U., Schulze J. P., Walz S. P., Lang U.: Evaluation of a collaborative volume rendering application in a distributed virtual environment. In *Proceedings of the workshop on Virtual Environments 2002 (EGVE '02)* (Barcelona, Spain, May 30-31 2002), The Eurographics Association, pp. 113–221.

[YO02] Yang H., Olson G. M.: Exploring collaborative navigation: the effect of perspectives on group performance. In *Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE '02)* (Bonn, Germany, September 30-October 2 2002), ACM, pp. 135–142.

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

A R T I C L E I N F O

Article history:
Received 1 September 2008
Accepted 13 January 2009

Keywords:
Collaborative virtual environments
Virtual reality
Asynchronous collaboration
Group dynamics

A B S T R A C T

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

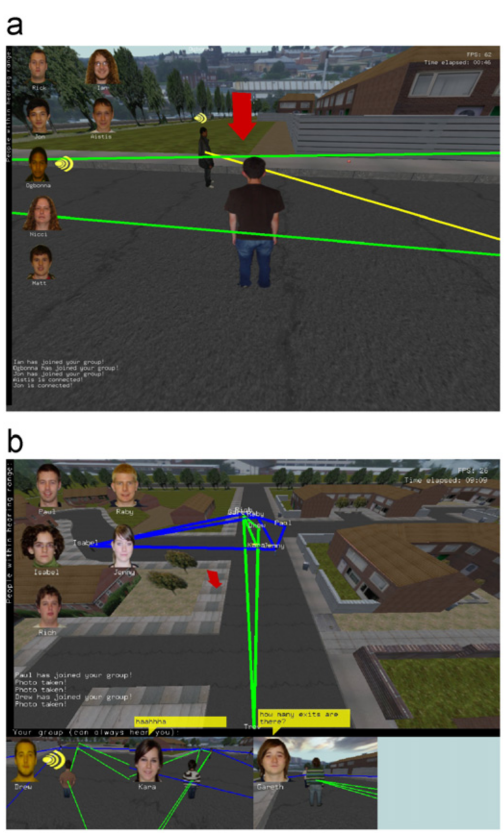


Fig. 1. Screenshots of the environment in the two conditions: teleport and multiple views. The graph metaphor, speech icon, teleporting arrow, and participants within hearing range can be seen in both figures. The views of fellow group members can be seen along the bottom of the screen in (b). (a) Teleport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

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.

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].

We consider a spoken or written utterance to be the basic unit of collaboration, a basic VT system would just contain what was said in that CVE, but nothing about who said them, what they were talking about, where they were in the CVE, or when. At the other extreme, a sophisticated VT system would allow you to travel through a virtual world, walking with people who had been there in the past, chipping in to their conversations as if they were still there, to the extent that an observer who came along later still would be unable to determine who were the original inhabitants vs. who was the impostor who has been added later?

Analysis of these examples highlights a rich complexity of possible functionality for VT. Therefore, the following sections present a framework of VT, and then describe the practicalities of implementation.

3.1. A framework for VT

Given that utterances are the basic building blocks of collaboration and communication in virtual worlds, then a key challenge for VT is determining how those utterances should be organized and associated. The primary methods for doing this are in terms of: (a) people, (b) time, (c) space and (d) topic. Each method has several levels (see Table 1), which can provide context to help us understand the meaning of what was said, influence where we choose to go next in the CVE, and help users control the number of utterances that are visible/audible at any given time so the VT system is scalable.

Adding people's identity to the utterances in a VT CVE allows users to discriminate everything that was said by a particular person, for example, someone who provided particularly insightful comments. Allowing people to choose their virtual appearance will have other effects on whose utterances a given user chooses to listen to, as found in real-time collaborative worlds [14].

Statistics terminology is adopted for the levels of time. At the nominal level, a future user would have no clue as to when, or in what order, different utterances were spoken. Ordinal information would allow utterances to be listened to in the sequence that they originally occurred, and the time interval (either absolute or rebased to when the speaker entered the environment) would allow sets of utterances that took place in quick succession to be distinguished from those that were separated by a lengthy delay.

Indicating the point in space where each utterance was spoken would help a future listener understand what was being talked about, and reduce the need for users in CVEs to devote much more effort to making the 'implicit explicit' than is the norm in real life [2,3]. Linking utterances by the path the speaker had taken would provide the listener with even more information about the things the speaker had seen and which led them to a particular conclusion.

Organizing and associating utterances can be done according to certain topics (or subtopics), for example defining whether greetings were due to users meeting or departing, whether idle chat was humorous or not, or which part of a task a given conversation was based on. Natural language processing (NLP) algorithms could be used to process utterances into topics, to which user-supplied quality ratings could be added by borrowing techniques from recommender systems and search engines.

Finally, there are many possible combinations of the above methods. For example, combining point (space) and topic (content) would allow the main items of interest in a given area to be quickly determined, adding interval (time) to identity (people) would allow the conversations of a group of people to be followed, and adding path (space) to interval/identity would help a future listener comprehend the bigger picture of a conversation that took place after a group of people had split up to explore an area and then regrouped to discuss their findings.

Table 1
Four methods for organizing/associating utterances.

Method	Level 1	Level 2	Level 3
(a) People	Anonymous	Identity	Appearance
(b) Time	Nominal	Ordinal	Interval
(c) Space	Amorphous	Point	Path
(d) Content	Undefined	Topic	Quality

Level 1 corresponds to a basic VT system, with Levels 2 and 3 providing ever richer possibilities for virtual time.

3.2. Implementing VT

For the evaluation described in Experiment 2, we implemented VT using level 3 utterance association for people (appearance) and level 2 association for time (ordinal), space (point) and content (topic) (see Table 1). Details of the implementation are as follows.

First, all of the utterances from two previous studies of synchronous teamwork in CVEs [4,5] were divided into blocks of communication and categorized as either task-specific or not. The latter were discarded to avoid cluttering the CVE with irrelevant utterances (e.g., idle chat). The task-specific blocks were classified using keywords from the 13 questions on the urban planning report that participants were asked to complete (see Section 4) and, although this was performed manually in the present study, it could have been done using NLP techniques.

The classification used a two level hierarchy, with the 13 questions (subtopics) grouped according to three topics (permeability, character, and safety and security) that were used on the urban planning report. The topics were rendered with different hues (yellow, cyan and magenta), using a different lightness for each subtopic. In addition, tags flashed when they were being played, and were visually caged in black stripes when they had been viewed. A color-coded checkbox was provided for users to choose which utterance subtopic(s) were displayed (see Fig. 2), allowing related comments to be identified even if they are separated in space and time.

The point where each utterance commenced was represented with a hemispherical visual object known as a tag, which put the conversations into context by showing where they took place. To reduce clutter, all utterances in a given block that were within line

Fig. 2. All of the utterance tags used in the VT study (Experiment 2). The 'Audio tag filters' was a list of checkboxes, shown in the top right hand corner of the screen, that allowed the utterances associated with each of the 13 questions to be toggled on/off. Utterances that a pair of participants had already listened to were visually caged in black stripes (see 'Exit points' on left-hand side of the image).

of sight of each other were represented by a single tag that was at the mean position of the individual utterance tags.

The system was tested using a pilot study and refined in response to participants' feedback. The main improvements were:

- Making tag selection explicit (instead of walking into a tag to play it, the users wanted to be able to select a tag with the mouse).
- Allowing users to pause/resume/stop tags, instead of always playing the whole of a tag.
- Providing more time for the task than was allowed in Experiment 1, because there were a lot of recorded utterances that the users wanted to watch and listen to.

Due to the exploratory nature of this work, it was difficult to generate meaningful hypotheses for the way participants would use the VT system and the changes it would make to their behavior. Experiment 2 was a first step in analyzing asynchronous collaborations in CVEs, and the results will help us make more informed design decisions and predictions in future work.

4. Experiment 1: real-time collaboration

The experiment used the context of urban planning, with participants asked to use a CVE to review the design of a new housing estate. Participants were run in two batches. In the first of these (the teleporting condition), participants had all the basic MGD functionality from [4], and new MGD functionality to provide awareness of who was talking, who was within hearing range and teleporting. Participants in the second batch were provided with multiple views (the multiple views condition), in addition to all the MGD functionality that was provided to the other batch of participants.

4.1. Method

The experiment took place in an undergraduate computing laboratory. Each participant was provided with a headset, and they were spread out across the laboratory so they could only communicate using audio and text communication from within the environment. Participants used two adjacent computers, one for the CVE and the other for the urban planning report write-up. The CVE application, environment and experimental procedure were the same as in [4].

4.1.1. Participants

All participants were undergraduate students from the School of Computing, who had not taken part in the previous study. Eight participants were recruited for each run, but one participant in the teleporting condition was unavailable on the day of the experiment. The remaining seven participants in the teleporting condition (6 men and 1 woman) had a mean age of 21.7 ($SD = 5.2$). The eight participants in the multiple views condition (5 men and 3 women) had a mean age of 21.8 ($SD = 4.1$).

All the participants volunteered for the experiment, gave informed consent and were paid an honorarium for their participation.

4.1.2. CVE application

The software application and 3D sound model are described in the previous study [4].

Distance attenuation was turned off for communications between members of the same group. This was clarified by displaying photographs of the faces of participants who would receive any transmitted audio. These faces were displayed on the HUD, and were added and removed appropriately as participants changed their position in the environment and switched groups. In addition, an icon was placed above a participant's avatar, and by the side of their face on the HUD, when they were talking.

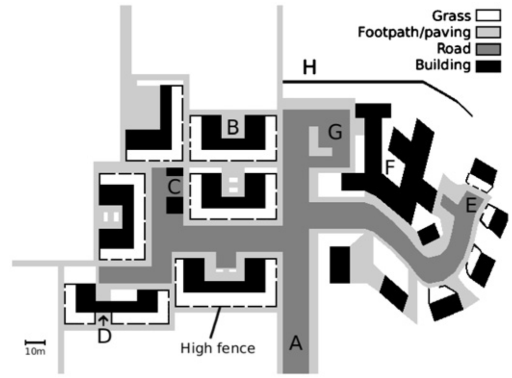


Fig. 3. A map of the estate. The estate had an entrance road in the middle (point *A*), which acted as a dividing line between two styles of building. On the left-hand side of the entrance road, there were brown-bricked terraced houses, which were mostly horse-shoe shapes creating partially enclosed private space (e.g. point *B*). The front gardens were bordered by high fences, and there were six garages in the road (*C*). There was an archway under one of the terraces (*D*). On the right-hand side of the entrance road there were red-bricked bungalows (single story buildings) along the edge of the curved road, with gardens bordered by low brick walls (e.g. *E*). There was a single-story care home for elderly people (*F*), with a car park to the left with space for six cars (*G*), and a hedge-row above it partly separating private land around the care home from public parkland (*H*).

4.1.3. Environment

The environment was a residential estate that was based on a real estate in Leeds. An annotated map of the estate is shown in Fig. 3.

All participants were represented in the environment with a photographic avatar (using four photos: front, back, left and right). Participants were given an over-the-shoulder perspective, with the option of switching to and from a bird's-eye view. An over-the-shoulder perspective meant that participants could see each other relative to their avatar, and be more aware of how others perceived them [15].

4.1.4. User interface

The participants used desktop workstations, and a two-handed control method, with one hand on the keyboard and the other hand on a 3-button mouse. By holding down appropriate arrow keys a participant could move forward/backward/left/right at 6 m/s, and heading and pitch could be changed by moving the mouse. This is a common gaming control method (e.g. [16]).

The 'Insert' key was used to take screenshots, the 'Home' key to toggle between over-the-shoulder and bird's-eye views, and holding down the 'Page Down' key allowed the participant to use voice communication.

Text communication was achieved by simply typing letters or numbers, which were transmitted the moment each was typed, appearing in a speech bubble above the participant's avatar. The text expired after approximately 10 seconds from the moment the enter key was pressed. Each participant was provided with a stereo headset for audio communication. The default recording and playback volumes were automatically set using a shell script.

The basic MGD functionality used three mouse buttons, and the 'Delete' key to move up one level in the group hierarchy. The display had a crosshair in the middle used for selection. Selecting an avatar with the left mouse button formed/joined a group. Selecting the avatar of a fellow group member with the right mouse button rapidly moved to their location and automatically followed them. Pressing the middle mouse button anywhere moved to the mean location of the group.

Holding down the numpad zero key released the mouse from controlling heading and pitch, and allowed it to control the position of the red teleporting arrow. Once the arrow was positioned in the desired location, a left mouse click teleported the participant there.

Participants in the multiple views condition could position the teleporting arrow over one of their group members views, and clicking the left mouse button would teleport them to that group member's location. By default, the participant's subsequent movements were tethered to that group member (the automatic following functionality in basic MGDs) but the participant could 'free' themselves simply by pressing a movement key.

The multiple views took up the bottom quarter of the screen. A limit was imposed of three views, each taking up a quarter of the horizontal space, with the remaining quarter reserved for displaying the faces of any other group members. These could be selected using the numpad zero key to release the mouse pointer. Selecting them showed their view in one of the existing viewports, swapping out the member whose view had been replaced.

4.1.5. Procedure

A 10 minute meeting was held with participants a few days before the experiment. They received a verbal explanation of the experiment, a single-sided A4 sheet containing extracts from UK urban planning guidelines and a consent form. They also had photos taken for their avatar during this time.

The experiment itself lasted one hour. At the start participants were provided with another copy of the urban planning guidelines sheet, an instruction sheet for using the CVE, an experiment schedule, and an electronic copy of an urban planning report which they had to complete during the experiment. The report contained the following questions, which participants were asked to illustrate using screenshots:

- *Question* 1, *Permeability*: (a) How many entrance and exit points are there around the estate? What are these for (i.e. cars or pedestrians)? (b) What reduces the speed/volume of traffic? (c) Are there suitable pedestrian routes around the environment? (d) Are the blocks small enough or do you have to walk too far before you reach a choice of direction?
- *Question* 2, *Character*: (a) Which parts of the environment follow the same pattern/building structure? (b) Find a part of the environment that is not consistent with the layout of the estate. (c) Is this acceptable or should it be changed? (d) Does the estate have character?
- *Question* 3, *Safety and security*: (a) Comment on the safety and security of the estate based on your own thoughts, the information in the guidelines and your discussion with other participants. (b) Find examples of where public and private space is clearly distinguished and where it is not. (c) Discuss which part(s) of the estate you think are least safe. (d) Can you find any blank walls that you think should be overlooked to improve the feeling of safety and help prevent graffiti? (e) Try to suggest some improvements with regard to the safety and security of the estate.

5. Results

There were two types of work that took place in the experiment: taskwork and teamwork [17]. Taskwork refers to the answers given in participants' reports, whereas data about teamwork were provided by the server's log of the movements, communication, and groups that participants formed.

The urban planning reports were marked like an exam. Participants names were on the reports, marking was not blind. An independent samples t-test showed no significant difference between the teleporting and multiple views conditions, $t(13) = 1.49, p = 0.16$. Participants in the teleport condition had a mean mark of 18.7 ($SD = 3.3$) out of 24, and 16.3 ($SD = 3.1$) in the multiple views condition. Our focus, however, was on how participants went about doing the task (i.e. the teamwork), and how different MGD functionality affected participants' behavior.

For each batch of participants, the spoken and text communication was transcripted and analyzed using a communication coding approach [18] to classify each utterance as one of the following:

(a) *Greetings* (e.g. 'Hey *M*!', 'Hi *G*!').
(b) *Functionality*—communication regarding the system and the groups (e.g. 'Think we need smaller groups than all of us!', 'You do realize that if you just press 'Home' you get a bird's-eye view and it's a lot easier to see!').
(c) *Environment*—discussion about the 3D world, but not in relation to the task (e.g. 'I swear you should be able to see uni from here.', 'I kind of might have figured out where the pictures were taken of, you know the Leeds skyscrapers ones.').
(d) *Task related* (e.g. 'Which part's the least safe?', 'I'd say where we're stood now, *J*.').
(e) *Idle chat* (e.g. 'Party at my flat. Come on, let's go!').

These data were analyzed in terms of the quantity of communication that took place, and where participants were relative to each other when they communicated. For comparison, data are provided from a previous study [4] when other participants had performed the same urban planning task either in a conventional CVE ('control' in Fig. 4) or with basic MGDs functionality (see Figs. 4 and 5). Note that the average group size in the basic MGDs, teleport and multiple view conditions was 3.5, 2.5, and 3.0, respectively.

The total number of utterances made by participants in the basic MGDs (data from [4]), teleport and multiple views conditions (data from the present study) was analyzed using a univariate analysis of variance (ANOVA). This showed that there

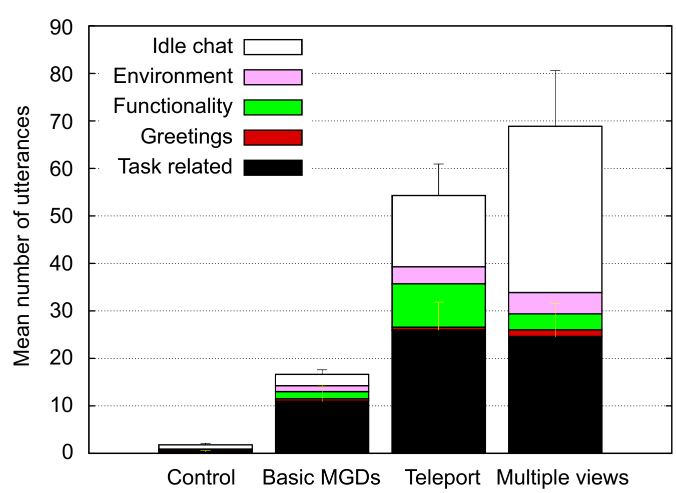


Fig. 4. Mean number of utterances made by the participants in each condition. The control and basic MGDs conditions are from [4]. The error bars are shown for task related utterances and idle chat.

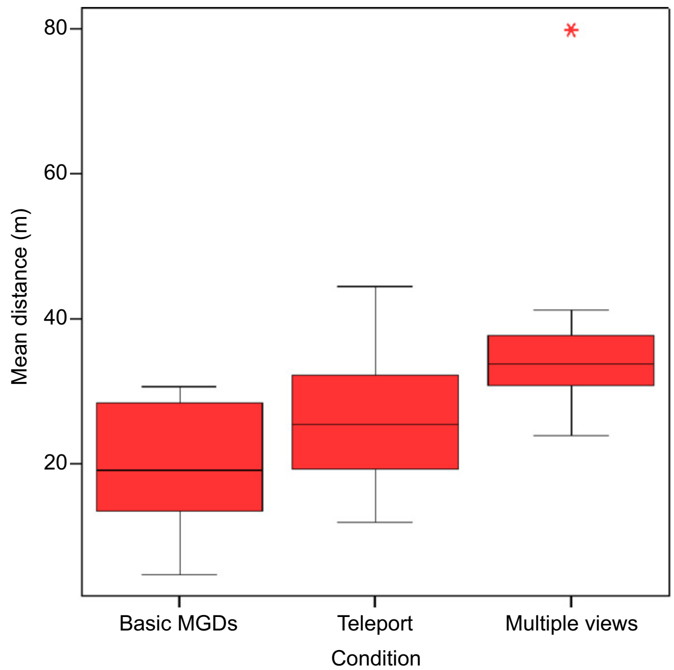


Fig. 5. Mean distance to the nearest group member at the time of each participant's utterances. The basic MGDs condition was from [4].

was a significant difference between the conditions, $F(2,20) = 3.91, p = 0.04$. Tukey HSD posthoc tests showed that the difference between basic MGDs and multiple views was significant ($p = 0.03$) but the other pairwise comparisons were not. The mean amount of communication increased by 226% from the basic MGDs to the teleport condition, and by another 27% from the teleport to the multiple views condition. Within this, task related communication increased by a factor of 2 from basic MGDs to the teleport and multiple views conditions, but this was not significant. Idle chat more than doubled from the teleport to the multiple views condition (see Fig. 4).

One of the limitations identified in our previous research was that participants tended to assemble in one place in the CVE before communicating, even though this was unnecessary with the basic MGDs functionality that was provided (see Introduction). To determine whether the new functionality provided in the present study overcame this limitation, each time a participant made an utterance the distance to their nearest group member was calculated, and the mean for each participant in the basic MGDs, teleport and multiple views conditions was analyzed using a univariate ANOVA. The two participants who did not speak at all during the experiment were excluded from the analysis, one was from the basic MGDs condition and the other was from the multiple views condition. The ANOVA showed that there was a significant difference between the conditions, $F(2,18) = 3.56$, $p = 0.05$. Tukey HSD posthoc tests showed that the difference between basic MGDs and multiple views was significant ($p = 0.04$) but the other pairwise comparisons were not (see Fig. 5).

6. Discussion

It is well known that in conventional CVEs users often have difficulty understanding the context of what each other is talking about (see problems 1 and 2 in Section 2). Our previous research into MGDs [4] showed how a group graph metaphor could help users find each other, since the graph 'tracked' participants and the nodes corresponded to avatars, with edges denoting group membership (see Fig. 1). One could find a group member by following a line from their avatar until they reached a node. In Experiment 1 of the present study, the teleporting and multiple views functionality took this a step further. It allowed participants to teleport directly to a group member of their choice by selecting the appropriate viewport, and providing participants with multiple views specifically tackled the problem of understanding another's perspective.

One of the places where the original MGDs techniques fell short of their goals was in facilitating communication when participants were spatially separated within the environment [4]. The fact that participants tended to collocate to communicate in the basic MGDs condition was a sign of inefficient groupwork—participants were either taking time to collocate when they wanted to communicate, or they were waiting until they were coincidentally collocated before they said anything.

The present study indicates that by providing feedback to the participants, they became more aware of how the system works and communicated across greater distances than in the basic MGDs condition (see Section 5). The interesting thing about this feedback from the system is it is not specifically new functionality in the sense of a new tool at the users' disposal, like teleporting and multiple views are (in the previous research it was possible for participants to communicate at a distance, because distance attenuation was turned off for within-group audio communication). Instead it boosts *awareness* of existing functionality. As Schroeder et al. reflect, do we improve usability 'by means of improving the systems and features of the environment, or by improving the users' awareness of their activities and settings?' [19, p. 666].

Finally, in previous research, participants communicated a great deal to overcome the lack of sensory information that CVEs provided [2,3]. By contrast, in the present study substantially more communication took place when extra sensory information was provided (e.g. awareness of who could hear you and who was speaking, and multiple views providing an 'extra pair of eyes'). This increase in communication was indicative of more teamwork taking place.

7. Experiment 2: VT collaboration

The second experiment used the same urban planning context and the same environment as Experiment 1. Participants were run in pairs, and had access to all the task-related conversations of 23 people who had previously done the same task in the environment (the 15 participants of Experiment 1, and the 8 MGDs participants from [4]). These previous conversations were embedded in the environment using the tags described in Section 3.2.

7.1. Method

The method was similar to that of Experiment 1. A total of 10 participants (5 pairs) took part. There were 7 men and 3 women, with a mean age of 22.2 ($SD = 3.3$). They had not taken part in any of the previous studies. Each pair communicated with each other and had access to the task-related conversations of 23 participants who had done the task in previous real-time experiments.

After pilot testing, the total time for Experiment 2 was increased to 90 minutes from the 60 minutes used in Experiment 1. The time in the training environment was extended from 15 to 30 minutes, and the time in the residential environment from 30 to 45 minutes. The final 15 minutes were allocated for a semi-structured interview.

The grouping interface and functionality were identical to that of the multiple views condition in Experiment 1, however, each pair was placed into a group together and could not join the groups of participants from the past. Participants could select a conversation tag by positioning the crosshair with the mouse and pressing the left mouse button. Participants could stop conversation tags by pressing the 'Escape' key, and pause/resume the playback with the 'F1' key.

7.2. Results

As in Experiment 1, the urban planning reports were marked like an exam, and participants had a mean mark of 17.9 out of 24 ($SD = 3.9$). An independent samples t-test showed no significant difference between the teleporting and multiple views conditions, $t(16) = 0.978, p = 0.343$. However, our main interest lay in how participants used VT and the effect it had on their behavior in the CVE. To investigate this, participants' communication, movement and tag usage were analyzed. Statistical comparisons were made with the multiple views condition from Experiment 1, whose interface was the same as the one used in Experiment 2 except for the VT functionality. In considering the findings, readers should bear in mind obvious differences between the experiments (especially the number of live participants at any given time), which could have affected the results.

Each participant's rate of communication was calculated by dividing the number of utterances they made by the time they spent working in the CVE (the difference between the time they first and last moved). This took account of the extra time allowed for Experiment 2 as a whole (45 vs. 30 minutes) and the fact that some participants remained 'in' the CVE (but not moving) while they finished writing their report.

Fig. 6 shows the mean rate of communication for task-related and non-task-related utterances for the multiple views (Experiment 1) and VT (Experiment 2) groups. An independent samples t-test was carried out on the task-related communication, and showed that the difference between the groups was significant, $t(16) = 3.258, p = 0.005$.

The distances that participants covered as they walked and teleported around the environment were calculated from the server log. The rate of travel was calculated by dividing the distance each participant traveled by the time they spent working in the CVE (calculated as above). Independent samples t-tests showed that the multiple views group walked significantly further in unit time than the VT group, $t(16) = 2.790,\ p = 0.013$ (Fig. 7).

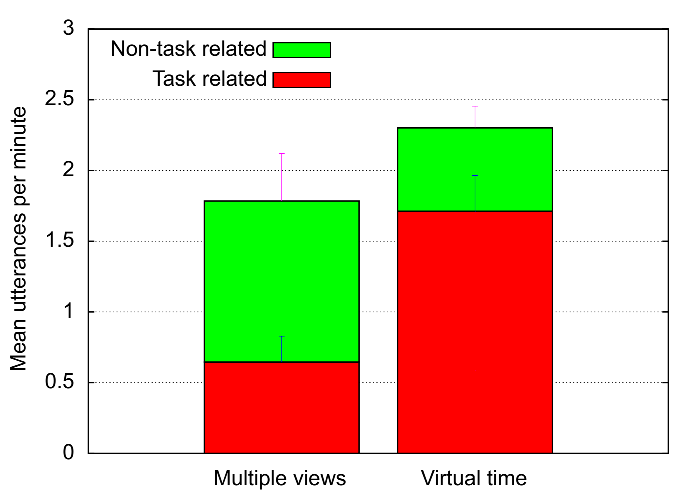


Fig. 6. The mean utterances per minute for the multiple views and virtual time conditions.

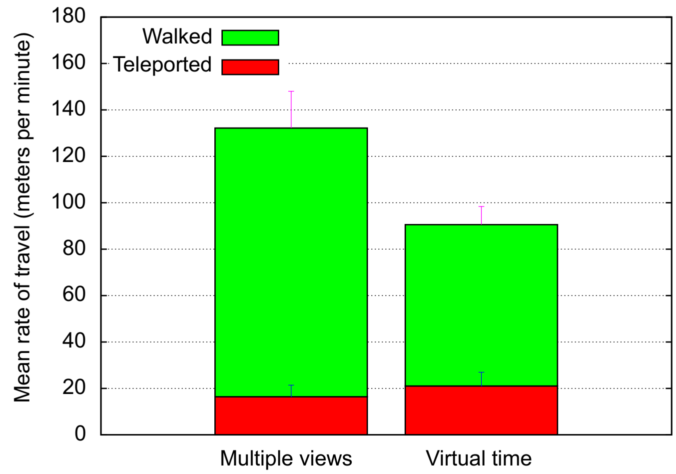


Fig. 7. The mean rate of travel for the multiple views and virtual time conditions.

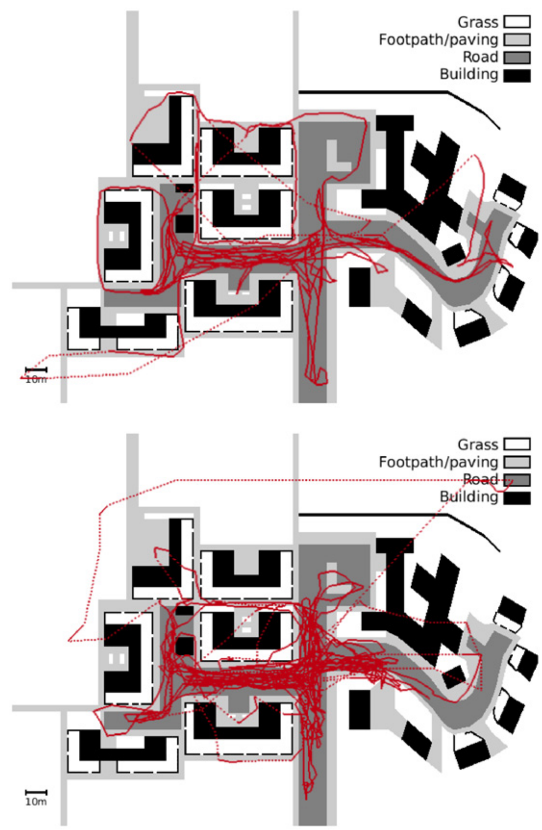


Fig. 8. The paths of two participants who had the median percentage distance teleported. The solid lines represent walking and the dotted lines represent teleporting.

The difference in rate of travel for teleporting was not significant, $t(16) = 0.578,\ p = 0.571$.

The paths of two participants were plotted as a qualitative analysis of the usage of teleporting. The two participants chosen had the median percentage distance teleported. The paths are shown in Figs. 8(a) and (b). The paths show examples of how teleporting was used: to cover large distances and to 'jump' over buildings.

There were a total of 67 tags containing 'VT' communication. All of participants' usage of the tags, and the tag filter menu, was recorded in the server log. Analysis of this log showed that pairs of participants typically selected one subtopic in the menu at a time,

Table 2
The mean number and SD of utterance tags in each subtopic that were played by each pair of participants.

Topic	Subtopic	Total tags	Mean tags played	SD	%
Permeability	Exit points	18	4.80	3.49	26.7
	Traffic	10	4.20	2.39	42.0
	Pedestrians	0	0.00	0.00	0.0
	Block size	4	2.00	1.58	50.0
Character	Buildings	3	1.40	0.89	46.7
	Inconsistent	5	0.80	0.84	16.0
	Acceptable?	0	0	0	0.0
	Character	4	2.00	1.41	50.0
Safety and security	Security	10	1.40	1.52	14.0
	Public/private	3	1.00	1.22	33.3
	Least safe	2	0.40	0.55	20.0
	Graffiti	4	0.80	1.30	20.0
	Improvements	4	1.20	1.30	30.0

Fig. 9. The environment with the utterance tags. The tags are colored based on how many pairs of participants played them: white = never played, blue = played by the minority of pairs, and red = played by the majority (3+).

allowing participants to focus on VT utterances that were relevant to the question being answered at a given time, and went through the tags in a logical order (subtopic by subtopic, matching the order in which the questions appeared in the urban planning report).

The mean number of tags played by each pair was 20.0 ($SD = 6.6$), and the breakdown by (sub)topic is presented in Table 2 (note: the playback was shared across the network, so both participants in a pair heard the same utterances). The distribution of the utterance tags and the frequency with which each was played is represented in Fig. 9.

7.3. Discussion

In this study, VT was implemented via a system of conversation tags (Sections 3.1 and 3.2) so participants could take advantage of the comments their predecessors had made. Participants used the environment in pairs, so each had one other real-time collaborator to communicate with and the conversations of 23 previous inhabitants to listen to. The results showed that the VT system led to a significant increase in task related communication between the 'live' pair when compared to the same interface without the VT (the multiple views condition). Furthermore, the results showed significantly less travel around the environment.

The reduced travel suggests that the points of interest were found by watching and listening to the conversations from the past: the usage results showed that on average a quarter of the conversation tags available for each subtopic were played. There is one drawback to this, however, as identified from the participant's comments in the semi-structured interviews. They were concerned about the quality of content of the tags, since they were using the information from past participants to perform the task. Assuming the past work was thorough and correct, VT provides a large pool of ideas to be shared from one group of participants to the next, thus focusing their conversation on the task. This highlights the importance of a measure of quality of content, level 3 in the VT framework, Section 3.1.

8. Conclusions

Our goal was to develop techniques to support synchronous and asynchronous collaborations in large-scale CVEs. For synchronous collaboration we identified problems in conventional real-time CVEs and built upon our existing MGDs functionality by adding teleporting and multiple view conditions. For asynchronous collaboration we presented a framework for VT and used the data from the real-time studies to implement VT, so live participants could benefit from their predecessors' comments.

The results from the real-time experiment (Experiment 1) showed that the awareness MGDs functionality produced a significant increase in communication, and an increase in the distance over which participants communicated, making participants behavior more efficient. In the VT experiment, participants listened to an average of 27% of their predecessors' conversations, spent more time talking about the task themselves and traveled significantly less. In other words, VT stimulated communication between the live participants.

Finally, future developments of VT systems could take many directions within the various levels of associations identified in Table 1. In particular, research is needed into the issue of quality of content of the conversation tags (level 3 of content). For example, tags could be user-generated, so that participants could explicitly request that particular conversations be recorded and tagged, a user-rating system would allow participants playing back the conversations to contribute to the quality control, and methods from recommender systems (e.g., Amazon's 'people who bought X also bought Y') could be incorporated.

Acknowledgment

The first author is funded by a Doctoral Training Grant Studentship from the School of Computing, University of Leeds.

References

[1] Johansen R. GroupWare: computer support for business teams. New York, NY, USA: The Free Press; 1988.
[2] Hindmarsh J, Fraser M, Heath C, Benford S, Greenhalgh C. Object-focused interaction in collaborative virtual environments. ACM Transactions on Computer–Human Interaction 2000;7(4):477–509.
[3] Ruddle RA, Savage JCD, Jones DM. Verbal communication during cooperative object manipulation. In: Proceedings of the 4th international conference on collaborative virtual environments (CVE '02), Bonn, Germany, September 30–October 2 2002. ACM. p. 120–7.
[4] Dodds TJ, Ruddle RA. Mobile group dynamics in large-scale collaborative virtual environments. In: Proceedings of IEEE virtual reality (VR '08), Reno, Nevada, USA, March 8–12 2008. p. 59–66.

[5] Dodds TJ, Ruddle RA. Using teleporting, awareness and multiple views to improve teamwork in collaborative virtual environments. In: Proceedings of the 14th eurographics symposium on virtual environments (EGVE '08), Eindhoven, The Netherlands, May 29–30 2008. p. 81–8.
[6] Pekkola S. Critical approach to 3D virtual realities for group work. In: Proceedings of the second nordic conference on human–computer interaction (NordCHI '02), Denmark, October 2002. ACM. p. 129–38.
[7] Yang H, Olson GM. Exploring collaborative navigation: the effect of perspectives on group performance. In: Proceedings of the 4th international conference on collaborative virtual environments (CVE '02), Bonn, Germany, September 30–October 2 2002. ACM. p. 135–42.
[8] Wössner U, Schulze JP, Walz SP, Lang U. Evaluation of a collaborative volume rendering application in a distributed virtual environment. In: Proceedings of the workshop on virtual environments 2002 (EGVE '02), Barcelona, Spain, May 30–31 2002. The Eurographics Association. p. 113–221.
[9] Sonnenwald DH, Whitton MC, Maglaughlin KL. Evaluating a scientific collaboratory: results of a controlled experiment. ACM Transactions on Computer–Human Interaction 2003;10(2):150–76.
[10] Ruddle RA, Howes A, Payne SJ, Jones DM. The effects of hyperlinks on navigation in virtual environments. International Journal of Human–Computer Studies 2000;53(4):551–81.
[11] Mackinlay JD, Card SK, Robertson GG. Rapid controlled movement through a virtual 3D workspace. Computer Graphics 1990;24(4):171–6.
[12] Benford S, Fraser M, Reynard G, Koleva B, Drozd A. Staging and evaluating public performances as an approach to CVE research. In: Proceedings of the 4th international conference on collaborative virtual environments (CVE '02), Bonn, Germany, September 30–October 2 2002. ACM. p. 80–7.
[13] Greenhalgh C, Flintham M, Purbrick J, Benford S. Applications of temporal links: recording and replaying virtual environments. In: Proceedings of IEEE virtual reality (VR '02), Orlando, FL, USA, March 24–28 2002. p. 101–8.
[14] Schroeder R. Social interaction in virtual environments: key issues, common themes, and a framework for research. In: Schroeder R, editor. The social life of avatars: presence and interaction in shared virtual environments. London: Springer; 2002. p. 1–18.
[15] Cheng L, Farnham S, Stone L. Lessons learned: building and deploying shared virtual environments. In: Schroeder R, editor. The social life of avatars: presence and interaction in shared virtual environments. London: Springer; 2002. p. 90–111.
[16] Brown B, Bell M. CSCW at play: 'There' as a collaborative virtual environment. In: Proceedings of the 2004 ACM conference on computer supported cooperative work (CSCW '04), Chicago, Illinois, USA, November 2004. ACM. p. 350–9.
[17] Baker K, Greenberg S, Gutwin C. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In: Proceedings of the 2002 ACM conference on computer supported cooperative work (CSCW '02), New Orleans, Louisiana, USA, November 16–20 2002. ACM. p. 96–105.
[18] Bowers CA, Jentsch F, Salas E, Braun CC. Analyzing communication sequences for team training needs assessment. Human Factors 1998;40(4):672–9.
[19] Schroeder R, Heldal I, Tromp J. The usability of collaborative virtual environments and methods for the analysis of interaction. In: Presence: teleoperators and virtual environments, vol. 15 (6). Cambridge, MA: MIT Press; 2006. p. 655–67.