

AUGMENTED REALITY FOR COLLABORATIVE EXPLORATION OF UNFAMILIAR ENVIRONMENTS

B. Bell and S. Feiner
Department of Computer Science
Columbia University
New York, NY 10027

EXTENDED ABSTRACT

Abstract. Understanding an unfamiliar environment can be a difficult and time-consuming task, yet one at which users can be particularly effective when it is accomplished collaboratively. We are investigating techniques that support communication among multiple users within a group to help them explore their surroundings together. Our focus is on determining how each user's interface is designed, based on their view of the environment and information received both from the environment and from other users. We build on our previous work on view management to assist in the placement of virtual information and to maintain a coherent user interface. We also use a situation-awareness aid, based on a head-controlled 3D World-in-Miniature, to display information on a head-worn display about objects or locations that might not be directly visible to the user.

1. INTRODUCTION

Augmented Reality (AR) [1] can potentially provide a powerful medium for collaborating in rich information spaces, allowing users to explore the surrounding environment, while still viewing and interacting with each other. However, as the amount of information that is available grows, it can become difficult to present effectively, whether on a mobile user's tracked head-worn display or on a stationary wall-sized display. There may be too much information to display overall, or even too much related to a specific portion of the environment. Some information might be vital for all users to know immediately, while other information might be relevant only to a specific user who is sufficiently close to a location with which it is associated. Information filtering is an important way to address these issues as a changing function of a user's task, situation, and other parameters [2]. However, changes in information filtering parameters can result in changes in display content—changes that are compounded with ones that result from changes in visibility as users and objects move relative to each other. To avoid disrupting the user's tasks, it is important for the user interface to remain coherent and informative throughout these changes. This is especially true in virtual reality (VR) and AR, in which users continually rely on their first-person views of the world to navigate, not just for information gathering and understanding.

We are applying view-management techniques to assist in the placement of virtual information and to maintain a coherent user interface. *View management* refers to the task of creating and preserving

desired visual relationships among virtual and physical objects in the user interface [3]. We accomplish this by allowing both the system and its users to manage certain graphical attributes of the objects in the user interface. For example, if we wish to avoid having less important objects occlude more important ones, our system could accomplish this by rendering less important potential occluders as transparent, or by controlling the size, shape, and position of virtual objects to prevent undesired occlusions. Note that the solution depends not only on the relative positions of users and physical objects, but also on the user's tasks and situation, which determine the relative importance of objects.

2. COLLABORATIVE VIEW MANAGEMENT

There are many ways that a collaborative system and its users can manage multiple views. Figure 1 shows how information can move between different entities in our system: users can send information to and receive information from any user in the environment, as well as the environment itself.

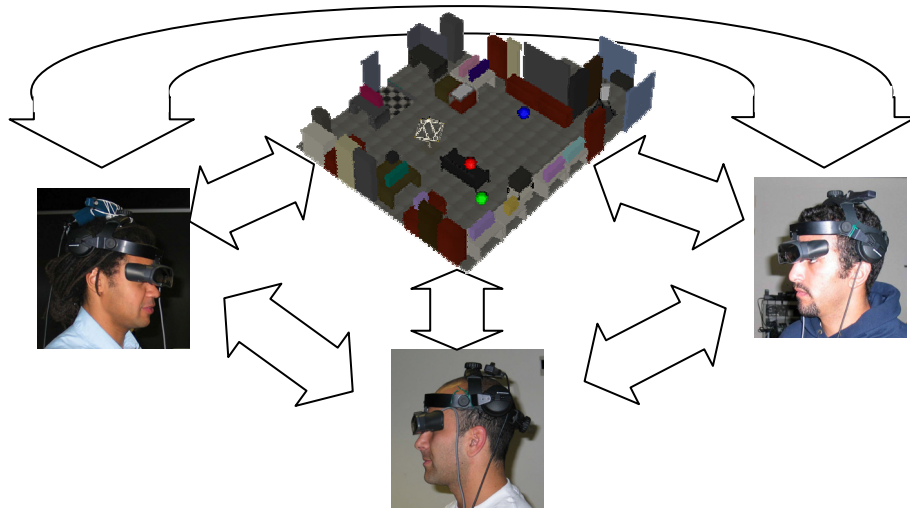


Figure 1. Each user sends and receives information to and from every other user in the environment and the environment itself.

2.1. Keeping Track of Important Objects

We may wish to keep track of certain objects (e.g., other team members), independent of whether they are being seen by us or by others. Figure 2(a) shows the view of a single user. Other users that are directly visible are labeled in the user's view of the real world; all remaining users are labeled in the

user's situation awareness aid, in which each is represented by a colored sphere. As users move into and out of view, the system automatically moves their labels between the real world and the aid, using animation to maintain coherence. Figure 2(b) includes labels for additional objects. In both figures, labels are laid out using the view-management algorithms of [3].



Figure 2. Viewing a collaborative augmented reality environment (imaged through a tracked video see-through display). (a) User labels only. (b) Additional objects labeled.

2.2. Monitoring Other Users' Views

We may wish to visualize what others are currently viewing. To make this possible, we have built on our earlier work on a situation-awareness aid [4], based on a head-controlled 3D World-in-Miniature (WIM) [5]. Presenting an overview of the environment in the aid could help the user get the big picture, while focusing on a particular object in the aid could provide details about it. Figure 3 shows a collaborative environment in which the virtual representations of real-world objects in the situation-awareness aid are colored based on the number of users that currently have that particular object in their field of view: the more users viewing an object, the brighter it becomes. This visualization might be useful to a team that needs to maximize coverage of an area being monitored, or to determine which objects are currently attracting the attention of others. In our implementation, each user's own computer determines the visibility of their environment, based on their position and orientation, and sends this information to any user that has requested it. A variation of this approach accumulates visibility information over time, and uses it to color an object to represent the number of different users that have seen it or the amount of time that it has been in view.



Figure 3. Situation-awareness aid being used to monitor what other collaborating users are viewing. Each object's brightness in the aid depends on how many users are viewing it. The user "Blaine" (magenta) is not visible in the real world; however, from the visibility information in the aid it is possible to tell that he is also looking in this direction.

2.3. Notifying Users of Other Users' Actions

Notification mechanisms are commonplace in current software systems; for example, informing us about incoming email, instant messaging, software updates, and bug fixes. We are exploring the use of notifications in our collaborative environment, in which a user might want to be notified whenever other users do certain things. For example, Figure 4 shows the result of a request for notification when at least two other users are viewing a display in the real world and the local user is not. In this case, the system highlights the projection screen in the situation-awareness aid, and labels it to point it out. (The annotation might also show what the other users are viewing on the screen.)



Figure 4. Notification rule that points out any display that two users are viewing and that is not in the local user's field of view. In this case, the Projection Screen fits the rule, resulting in the display of a related label in the aid.

3. IMPLEMENTATION

The examples in this extended abstract are based primarily on computing visibility information for all collaborating users in the environment. We use a rule-based approach to specify how information should be computed and communicated to each of the users. Each user has a set of data tables that

represent their situation. Figure 5 shows simplified versions of some of the data tables for the local user whose view is seen in Figure 4. The *User* table has a row (i.e., record) for each user, with the *Is Local* flag set to *TRUE* only for the user who owns the table. The *Visibility* table encodes which objects are seen by which users. The *Objects* table indicates the number of users looking at each object and whether the local user is among them.

Certain rows are distributed from each user's tables. For example, each user distributes a row in the *Visibility* table for each object to every other user in the environment, as well as incremental updates to the *Visible* column that are based on the distributing user's own visibility calculations. (Visibility calculations for a user are currently done only on that user's computer for performance reasons.) For the *User* table, only the user's own row is distributed (without its *Is Local* flag).

The *Objects* table does not get distributed at all: it keeps track of all of an object's information locally, and is recomputed for each user. Two fields in the *Objects* table (*Number of Lookers* and *Is Local Looking*) can be computed from the two other tables (*Visibility* and *User*). Therefore, we set up computations that occur in the *Visibility* table, based on inputs from the *Visibility* table itself and the adjacent *User* table. A local computation increments *Number of Lookers* by one when an object becomes visible to any user, and decrements it by one when it stops being visible to any user. Therefore, as updates to the *Visible* column are received by remote users, *Number of Lookers* will also be updated. Each *Is Local Looking* entry for an object in the *Objects* table is a copy of the *Visible* entry for the local user's record for that object in the *Visibility* table.

User		Visibility			Objects		
User	Is Local	User	Object	Visible	Object	Number of Lookers	Is Local Looking
Blaine	TRUE	Blaine	Screen	FALSE	Screen	2	FALSE
Eddie	FALSE	Eddie	Screen	TRUE	Cabinet	0	FALSE
Benko	FALSE	Benko	Screen	TRUE	Table	1	TRUE
Alex	FALSE	Alex	Screen	FALSE			
		Blaine	Table	TRUE			
		Eddie	Cabinet	FALSE			
		...	...	...			

Figure 5. Data tables that represent information used to develop rules. Incremental updates to the *Visible* column in the *Visibility* table are distributed to other users throughout each example. The *Number of Lookers* and *Is Local Looking* fields are computed values, based on information in the *User* and *Visibility* tables. In each example, different rules use this information to control program functionality.

In the example in Figure 4, there is also a computation that determines whether the two computed fields, *Number of Lookers* and *Is Local Looking*, match the rule. If they do, then the rule sets a Boolean value that fires all the computations needed to place a label relative to the virtual representation of that object.

4. CONCLUSIONS AND FUTURE WORK

We have presented several examples of how multiple users' views can be controlled in a collaborative AR environment, addressing some of the wide spectrum of ways in which people can work together.

Implicit in much of what we have described is the importance of security. Who should be able to determine what another person sees and how they can interact with it? Or what information about a user's interactions may be made available to others? To answer some of these questions, we will need to categorize the relationships between users. Conventional computer security mechanisms support separate read, write, and execute access for files for relatively static classes of users, such as owners, groups of users, and everyone else. In contrast, collaborative relationships among users appear to be comparatively complex and continuously changing. For example, a user might selectively grant another user the right to point out something to them only when both are within a certain range or are visible to each other. The system should be able to handle such rules easily and efficiently.

Software design and integration is also a difficult task in a distributed environment. Currently, peer-to-peer systems are used in relatively simple applications such as file sharing and distributed computing. We are currently designing and extending our rule-based system to address more collaborative behavior in a variety of AR tasks.

4. ACKNOWLEDGMENTS

This work was supported in part by ONR Contracts N00014-99-1-0249, N00014-99-1-0394, and N00014-99-1-0683; NSF Grant IIS-00-82961; NLM Contract R01 LM06593-01; an IBM Graduate Research Fellowship to Blaine Bell; and gifts from Mitsubishi Electric Research Labs and Microsoft.

5. REFERENCES

1. Azuma, R.T., A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 1997, 6(4): pp. 355–385.
2. Julier, S., M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, T. Höllerer, and S. Sestito. Information Filtering for Mobile Augmented Reality. in *Proc. ISAR '00 (Int. Symposium on Augmented Reality)*. 2000. Munich, Germany, pp. 3–11.
3. Bell, B., S. Feiner, and T. Höllerer. View Management for Virtual and Augmented Reality. in *Proc. UIST '01*. 2001. Orlando, FL, pp. 101–110.
4. Bell, B., T. Höllerer, and S. Feiner. An Annotated Situation-Awareness Aid for Augmented Reality. in *Proc. UIST '02*. 2002. Paris, France, pp. 213–216.
5. Stoakley, R., M. Conway, and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. in *Proc. CHI '95*. 1995. Denver, CO, pp. 265–272.