Information Filtering for Mobile Augmented Reality*

Simon Julier, Marco Lanzagorta, Yohan Baillot and Dennis Brown†

July 2, 2002

Introduction

Augmented Reality (AR) has the potential to revolutionise the way in which information is delivered to a user. By tracking the user's position and orientation, complicated spatial information can be directly registered to the real world in the context where it applies. We are focussing our research on the problem of developing mobile augmented reality systems which can be worn by an individual user operating in a large, complicated environment such as a city. Virtual sign posts can, for example, announce the name of anonymous streets. Hidden infrastructure such as sewer or gas lines can be shown beneath a road surface. However, an urban environment is extremely complicated: it is populated by large numbers of buildings, each of which can have numerous facts stored about it. Therefore, it is very easy to inflict the user with *information overload*. This problem is illustrated in Figure 1 which shows a screen capture from our mobile AR system[1]. The purpose of this application is simple: the system is trying to guide a user to an office in a small building. The application should start by guiding the user to the correct building, then to the correct entrance, and finally to the correct office. Figure 1 shows what happens when the system draws all the environmental data. The display includes both relevant information (such as the name and location of the building and the target office) and irrelevant information (a detailed geometric model of the exterior of the building, the interior of the building, and all other data which lies within the view frustum but is behind the foreground building). As can be seen, the display is extremely complicated, confusing and uninformative.

Figure 1: Showing all available data leads to clutter and confusion.

To overcome these problems, we have begun to develop algorithms for information filtering. These tools automatically restrict the information which is displayed to minimise problems of information overload. Although the algorithms are being developed in the context of mobile augmented reality, they are drawn from several research areas and we believe that the basic approach is applicable in many other problem domains.

Information Filtering Approaches

Physically Based Methods

The simplest way to filter information is to use information about the physical infrastructure of the environ-

*Portions of this paper first appeared in [1]

†S. Julier, Y. Baillot and D. Brown are with ITT AES / Virtual Reality Laboratory, Naval Research Laboratory, Washington DC. M. Lanzagorta is with Scientific and Engineering Solutions.

[1]All the pictures for the AR system in this paper were captured by mixing the output of our AR system together with data from a video camera. The low quality of the images is due to limitations with the current camera and video mixer configuration. If this paper is accepted, we shall obtain better images.

Figure 2: Distance-based is not sufficiently discriminating. Much irrelevant data is displayed.

ment. In particular, it is possible to use *distance-based* and *visiblity-based* filtering. Distance-based filters threshold an object's visibility purely on the basis of its distance from the user. If the distance exceeds some threshold d, the object is not shown to the user. Many graphics APIs generalise this concept through the introduction of a *level of detail*: as the distance increases, progressively simpler models are used. Visibility-based filters determine whether an object is visible to the user and, if so, augments the visible part. This has the advantage that much of the superfluous information behind the target building in Figure 1 is eliminated.

However, such simple strategies are unsatisfactory because importance is not simply a function of distance or visibilty from a user. The limitation of distance-based filtering is shown in Figure 2: the visibility distance d has been manually adjusted so that only the building which contains the office is visible. However, to ensure that the target office is visible, it is necessary to show a significant amount of building infrastructure and other irrelevant information. Visibility-only filtering undermines the important capability of providing a user with "X-ray vision" and be able to see information about objects which aren't visible. Furthermore, it still does not identify important information. In Figure 1 all of the objects on the front of the building would still be annotated.

Visibility Filtering

Spatial Model of Interaction

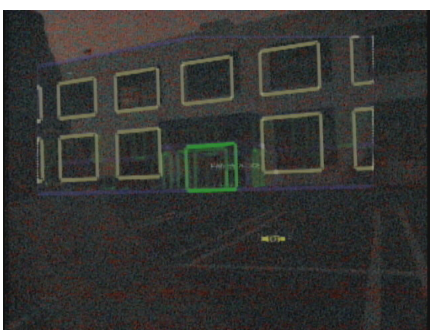
A more sophisticated version of distance-based filtering is the *spatial model of interaction* [2]. The spatial model was first developed to consider the problems of awareness and interaction in multi-user virtual environments, where awareness can be used to determine whether or not an object is visible to, or capable of interaction with, another object. In this model, each object (e.g., a user), is surrounded by a *focus*, specific to a medium (e.g., graphics or sound), which defines the part of the environment of which the object is aware in that medium. Each object in the environment also has a medium-specific *nimbus*, which demarcates the space within which other objects can be aware of that object. If the focus and nimbus intersect, the two objects can interact with one another.

The spatial model is a superset of simple visibility based filtering. By allowing objects focuses and nimbuses to be expanded, it provides further distance related information. The spatial model has the advantage that it allows different objects to be demarcated at different ranges. Furthermore, it can leverage efficient collision detection algorithms such as the Oriented Bounding Box Tree described in [3]. Figure 3(a) shows the results when the user is far away. The focus on the building and the entrance has been extended and therefore, they are the only objects which are visible. However, because the focus and nimbus are fixed, as the user moves closer, the user automatically sees more (irrelevant) data, as shown in Figure 3(b).

Rule-Based Filtering

Several researchers have addressed the problem of filtering through the use of inference engines and rule-bases. These are the most general form of information filtering algorithm. Arbitrary relationships can be specified, maintained and adjusted as a user's context and goals change. KARMA [4], for example, used a rule-based approach to select relevant information to assist a user performing a maintenance and repair task. The user's position and orientation, inter-object occlusion relationships, and the role that the objects play in a specific task to be accomplished by the user, all determine whether and how objects should be displayed, highlighted, and labeled on a tracked, see-

(a) At a distance, the spatial model can be used to discriminate between only the most important information by expanding the nimbus on far away objects.



(b) However, as a user draws closer, their focus intersects with the nimbus of all objects, irrespective of their relevance.

Figure 3: The Spatial Model of Interaction provides partial functionality required by an information filtering system.

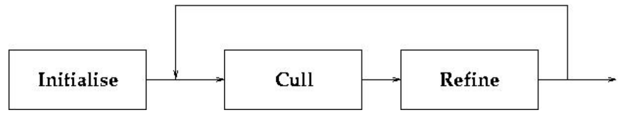


Figure 4: Block diagram of the filtering algorithm.

through, head-worn display.

However, the problem with this approach is its potential scalability concerns. The database of the examples shown in this paper includes 30 buildings and over 740 distinct objects, most of which are related to distant buildings which are simply not relevant to the current user's task. Applying potentially computationally expensive, high order decision logic to even such a simple example has the potential to impose a substantial computational burden. When the system is to be applied to a large environment such as a city, the computational costs could become prohibitive.

Hybrid Information Filtering System

From the previous discussion, it is clear that the most general form of information filtering is to use a rule-base. However, as explained above, it has significant computational concerns. The spatial model of interaction, to a first order approximation, is capable of performing the initial filtering which is required. Furthermore, it can leverage efficient collision-detection algorithms. Therefore, our algorithm is a hybrid of these approaches, and consists of the four stages which are shown in Figure 4 [1]:

1. **Initialize.** Given knowledge of the user's objectives and goals, calculate the user's focus and the nimbus for each object. This calculation is carried out whenever an object's property changes or the user's objective changes.

2. **Cull.** Use the spatial model of interaction to eliminate all objects whose nimbi do not intersect with the user's focus.

3. **Refine.** Apply higher order decision logic.

Stages 2 and 3 are performed periodically whenever the user's position and/or orientation has changed. Our current implementation of Stage 2 only uses the intersection of the focus and nimbus. However, other operations (such as visibility determination) could be incorporated as well.

To implement this algorithm, it is necessary to represent the user's objectives and goals, the relevance of objects to those goals, and provide a mechanism for calculating the focus and nimbus. We encode the notion of objectives and goals through the use of *objective* and *subjective* states which are assigned to each object and each user.

Objective properties are the same for all users, irrespective of the tasks which that user is carrying out. Such properties include the object's classification (for example whether it is a building or an underground pipe), its location, its size and its shape. This can be extended by noting that many types of objects have an *impact zone* — an extended region over which an object has a direct physical impact. A wireless networking system such as the WaveLAN, for example, is effective over a finite distance. This region can be represented as a sphere whose radius equals the maximum reliable transmission range. Conversely, a more accurate representation could take account of the masking and multi-path effects of buildings and terrain through modeling the impact zone as a series of interconnected volumes. Because of their differing physical properties, different media can have different impact zones.

Subjective properties attempt to encapsulate the domain-specific knowledge of how a particular object relates to a particular task for a particular user. Therefore, they vary between users and depend on the user's task and context. We represent this data using an *importance vector*. The importance vector stores the relevance of an object with respect to a set of domain-specific and user-scenario specific criteria. For example, if a user is following a route to a particular office, only that office and route information which leads to it is important — all other information is less important.

The objective–subjective property framework can be applied to model the state of each user. Each user has their own objective properties (such as position and orientation) and subjective properties (which refer directly to the user's current tasks). Analogous to the importance vector we define the *task vector* which stores the relevance of a task to the user's current activities. The use of a vector means that a user can carry out multiple tasks simultaneously and, by assigning weights to those tasks, different priorities can be assigned. For example, at a certain time a user might be given a task to follow a route between two points. However, the user is also concerned that (s)he does not enter an unsafe environment. Therefore, two tasks — route following and avoiding unsafe areas — run concurrently. The task vector is supplemented by additional ancillary information. In the route following task, the system needs to store the way points and the final destination of the route.

Example

The scenario is that a mobile user will be given directions to the location of Simon's Office. The system is illustrated in Figure 5, which shows the output of the system in three separate locations[2].

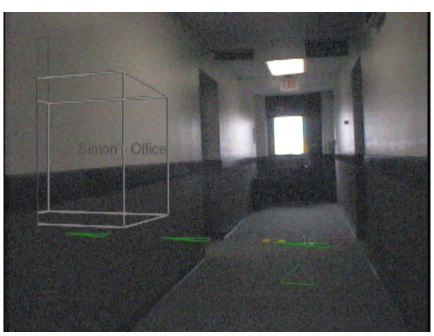
Figure 5(a), taken from the same position as that used in Figure 3(b) shows that the second stage of the filter eliminates all superfluous data not relevant to the route following task. Therefore, only the entrance to the building is displayed. Figure 5(b) is taken inside the building. A route has appeared, directing the user towards the office. Due to the action of the spatial model, only a subset of the route is shown at any given time to avoid confusing the user. In Figure 5(c), the user draws close to the final destination. The display shows a final turn to the left (potentially confusing in Figure 5(b)) and the final destination office.

Figure 5(b) shows a limitation with our current implementation. The blue rectangle to the left of the image is actually the front of the target building. This is a route-related object whose nimbus extends inside the building and therefore the filter determines it is relevant to the user. There are a number of ways to eliminate this artifact including the use of visibility information (in stage 3 of the filter), or redefining the task with a finer granularity. For example, the task could be decomposed into two tasks of entering the correct building and traversing to the correct office within that building.

[2]It should be noted that, to date, tracking systems which operate indoors, outdoors and could be deployed over the area of a building are still not available. For the purpose of this article, we assume that such tracking systems exist. For a review of current work in tracking systems, see the upcoming IEEE Computer Graphics and Applications special issue on tracking.

(a) View from the door, same as in Figure 3(b). Only the building and the correct entrance are annotated.

(b) View along corridor inside building. A route leads towards the final destination.



(c) As the user draws near the final destination, the destination office is shown as well as a final turn in the route.

Figure 5: Sequence from example. See text for a description.

Conclusions

In this paper we have discussed information filtering algorithms particularly tailored for the needs of mobile augmented reality systems. We have presented a hybrid system which allows the use of arbitrarily complicated decision models but, at the same time, can leverage spatial operators to significantly reduce scaling.

However, the work described in this paper only addresses the first of several stages required to build informative user interfaces. First, it is necessary to manintain visual constraints between the objects to be annotated and the annotations themselves. Blaine et al. refer to the maintenance of these constraints as *view management* and demonstrate algorithms which automatically size and position virtual labels such that the labels do not overlap one another or the objects which they are augmenting [5]. Second, it is unlikely that pixel-level registration can be achieved with wearable tracking systems. MacIntyre et al. have begun to develop algorithms to quantify registration errors to dynamically adjust augmentation to minimize potential ambiguities [6]. Both of these extensions introduce a coupling between objects which are filtered out and those which are not. Our current work is extending the filtering algorithm to explore these interdependencies.

References

[1] S. Julier, M. Lanzagorta, S. Sestito, L. Rosenblum, T. Höllerer and S. Feiner, "Information Filtering for Mobile Augmented Reality," in *Proceedings of the IEEE 2000 International Symposium on Augmented Reality, Germany*, IEEE, October 2000.

[2] S. Benford and L. Fahlén, "A Spatial Model of Interaction in Large Virtual Environments," in *Proceedings of ECSCW '93*, (Milan, Italy), September 1993.

[3] S. Gottschalk, M. C. Lin and D. Manocha, "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 171–180, 1996.

[4] S. Feiner, B. MacIntyre and D. Seligmann, "Knowledge-based augmented reality," *Commu-*

nications of the ACM, vol. 36, pp. 52–62, July 1993.

[5] B. Bell, S. Feiner and T. Höllerer, "View management for virtual and augmented reality," in *Proc. ACM UIST 2001 (Symp. on User Interface Software and Technology)*, pp. 101–110, ACM Press, 2001.

[6] B. MacIntyre, E. Coelho and S. Julier, "Estimating and adapting to registration errors in augmented reality systems," in *Proc. IEEE Conferece on Virtual Reality*, (Orlando, FL, USA), IEEE Press, March 2002.