Information Filtering for Mobile Augmented Reality

Simon Julier
Marco Lanzagorta
Yohan Baillot
Lawrence Rosenblum*
Advanced Information Technology
Naval Research Laboratory
Washington DC
{julier,lnzgrt,baillot,rosenblu}@ait.nrl.navy.mil

Steven Feiner
Tobias Höllerer
Dept. of Computer Science
Columbia University
New York, NY 10027
{htobias,feiner}@cs.columbia.edu

Sabrina Sestito
Defence Science and Technology Organisation
506 Lorimer St.,
Port Melbourne 3207
AUSTRALIA
sabrina.sestito@dsto.defence.gov.au

Abstract

Augmented reality is a potentially powerful paradigm for annotating the environment with computer-generated material. These benefits will be even greater when augmented reality systems become mobile and wearable. However, to minimize the problem of clutter and maximize the effectiveness of the display, algorithms must be developed to select only the most important information for the user. In this paper we describe a region-based information filtering algorithm. The algorithm takes account of the state of the user (location and intent) and the state of individual objects about which information can be presented. It can dynamically respond to changes in the environment and the user's state. We also describe how simple temporal, distance and angle cues can be used to refine the transitions between different information sets.

1 Introduction

Augmented reality (AR) integrates virtual information with the user's physical environment. Graphics-based AR can provide a user with a "heads up display" in which computer graphics is spatially registered with, and overlaid on, geographic locations and real objects. Experimental AR systems have been demonstrated in a range of potential applications, from aircraft manufacture [7] to image guided-surgery [11], and from maintenance and repair [9, 13] to building construction [23]. Improvements in portable computing hardware, position and orientation trackers, and see-through displays promise to make wearable AR a commercial reality this decade.

If a graphics-based AR system is to be effective, care must be taken to ensure that its display is not cluttered with too much information. This problem is illustrated in Figure 3(a), which is imaged through the see-through head-worn display of an experimental mobile AR system currently under development at the Naval Research Laboratory [3]. The system presents all the data that it has about the environment. The resulting display is highly cluttered and many of the labels and wire frame diagrams obscure both one another and the environment. Such clutter can undermine the effectiveness of an AR display [21].

One way to address this problem is through *information filtering*. Information filtering means culling the information that can potentially be displayed by identifying and prioritizing the information that is relevant to a user at a given point in time. In the case of AR or other situated user interfaces [15, 8] that take into account the user's location, information can be classified based on the user's physical context, as well as on their current tastes and objectives. Information filtering is a key component of *environment man-*

*S. Julier and Y. Baillot are with ITT AES. M. Lanzagorta is with Scientific and Engineering Solutions.

agement [17], a term that we have previously used to refer to the task of managing the large number of displays, interaction devices, and virtual objects with which a user will interact in a world populated with ubiquitous mobile and wearable computers.

In this paper, we consider the problem of information filtering for mobile augmented reality and describe an information filtering algorithm that we have developed. We begin by describing our application scenario in Section 2 and previous work in Section 3. Next, we discuss our information filtering mechanism in Section 4, and our current implementation of the algorithm in Section 5. Finally, Section 6 presents our conclusions and future work that we intend to explore.

2 Application Scenario

Our goal is to develop software systems and interaction techniques to support multiple, mobile, collaborating users with wearable AR systems. These users would interact with other users of stationary VR, AR, and desktop systems. We consider the following basic application scenario [14]:

- Multiple users with mobile AR systems are free to roam through an urban environment. Each user performs one or more tasks (such as "follow a route between two specified points"), which can be acted upon sequentially or concurrently.
- The AR systems help users accomplish their tasks by providing them with relevant information about their environment. For example, this might include names and other properties of buildings and infrastructure that may or may not be directly visible from a user's current location.
- Users can interact with the information presented to them; for example, by creating annotations that can be attached to locations or objects.
- Collaboration among mobile users can be achieved through sharing information; for example, by mobile users exchanging annotations.
- A supervisory "base station" (e.g., a stationary multi-user virtual environment system) oversees the actions of the mobile users. Base station users receive information from mobile users and can send them additional information about the environment and their tasks.

This is an extremely general scenario that is relevant to a wide range of applications, including field maintenance or sales, law enforcement, the military, utility and emergency services, and even tourism.

Our groups have developed two mobile AR systems, one of which is shown in Figure 1. Each system is composed of 6DOF trackers (an Ashtech GG Surveyor real-time–kinematic GPS for position, an InterSense IS300Pro for orientation), a see-through head-worn display (Sony LDI-D100B Glasstron), a wireless network (a FreeWave radio modem), and a wearable computer with 3D hardware graphics acceleration (using either ABX or PC104 form factor [3]). Our current test datasets, of approximately 30 buildings, include about 150 objects. The types of objects include buildings, windows, doors and tunnels. Future models are likely to contain at least an order of magnitude more objects.

Given the number and the density of the objects, information filtering is vital to prevent cluttering the display. An informal domain analysis of our application scenario suggested to us that the filtering mechanism should take into account the following properties:

- Users will perform a broad range of tasks, from maintaining general situational awareness of their environment, to searching for specific objects, to attending to a specific set of objects involved in an activity.
- Any object, of any type, at any point in time, can become sufficiently important that it must be able to pass the filtering criteria.
- Certain objects are important to all users at all times.
- Certain objects are important to all users whenever they are performing a particular task.
- Some objects (such as the way points that define a route) are only important to the activities of a particular user.
- All things being equal, the amount of information shown to a user about an object is inversely proportional to the distance of that object from the user. For example, at a sufficient distance, the only information that might be shown about a building would be a text label. As the user approaches the building, more information might appear (e.g., locations of main entrances and exits). Finally, at very close distance, the user might be shown information about the physical contents of the building (e.g., a floor plan).

3 Previous Work

Filtering crowded information displays to prevent clutter and improve human performance or rendering performance has long been recognized as a potential problem for information display systems [19]. Control of the level of detail

Figure 1. Prototype mobile AR system.

at which an object is viewed (in the limiting case, determining whether it is even visible), based strictly on its distance from the viewpoint or the size of its projection, is supported in many current 3D modeling languages, and was originally developed for early 3D vector graphics systems [22]. For example, in VRML 97 [6] and Java3D [20], different versions of an object's geometry can each be associated with a specific distance range between the viewpoint and object.

Fish-eye views [12] provide a general approach to filtering object display based on a combination of (spatial or conceptual) distance of an object from one or more focal points and some measure of an object's a priori importance.

The spatial model of interaction [4] treats 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. In the general spatial model of interaction, each object has a different focus and nimbus for each medium supported by the system, nimbi and foci can be of arbitrary size and shape (e.g., asymmetric or disjoint) and may be discrete or continuous, and the awareness that object $\mathbf{A}$ has of object $\mathbf{B}$ in a particular medium is some function of $\mathbf{A}$'s focus and $\mathbf{B}$'s nimbus in that medium. Specific examples of the model have been implemented in the MASSIVE and DIVE systems [5], which take different approaches to computing awareness. For example, in DIVE, awareness is a binary function, where $\mathbf{A}$ is aware of $\mathbf{B}$ if $\mathbf{A}$'s focus overlaps with $\mathbf{B}$'s nimbus. In contrast, in MASSIVE, foci and nimbi are scalar fields radiating from point-sized objects, focus and nimbus values are sampled at each object's position, and $\mathbf{A}$'s level of awareness of $\mathbf{B}$ is the product of $\mathbf{B}$'s value in $\mathbf{A}$'s focus and $\mathbf{A}$'s value in $\mathbf{B}$'s nimbus.

Several researchers have addressed the problem of filtering overlaid information for AR. KARMA [9] uses 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-through, head-worn display. Although implemented in a stand-alone VRML browser, rather than in an AR system, InfoLOD [18] determines whether and how to label buildings in a virtual cityscape. In InfoLOD, information is associated with specific sides of cuboidal buildings; visibility decisions are based on a building's distance from the viewer and on the building's orientation relative to the viewer, making it possible to to treat information associated with different sides differently.

Some of the approaches that we have surveyed here are just high-level techniques: the distance-based level-of-detail support of VRML 97 and Java3D, fisheye views, and the abstract spatial model of interaction. Therefore, a number of implementation decisions would need to be made to instantiate any of these. The remaining approaches correspond to actual implementations: KARMA's rule-based technique, the specific versions of the spatial model of interaction used in MASSIVE and DIVE, and InfoLOD. Of these, KARMA's rule set is designed to accommodate physical tasks that require the display of a relatively small number of objects that are directly related to the task to be performed, and does not address the wider range of tasks required of our users. InfoLOD does not take into account the user's current task or object importance metrics at all, while MASSIVE and DIVE can through varying the size and shape of foci and nimbi. However, none of these implementations of the spatial model of interaction deal with the specifics of how object importance metrics and task information can be incorporated, or are designed for the AR information overlay tasks in which we are interested.

4 A Region-Based Information Filter

4.1 The Use of Subjective and Objective Properties

The information filtering system should, at any given time, only show information which is important to the user. However, the importance of a piece of information depends on the user's current context. More specifically, we assume that each user is assigned a series of tasks. Each task requires that the user interacts with a set of objects in certain ways. To model these effects, we assume that each user can interact with objects through a set of *medium* and that each user and each object possesses both *objective* and *subjective* properties.

The concept of a medium, defined in the spatial model [4], describes the way in which a user interacts with an object. The original implementations in DIVE and MASSIVE assumed that only three types of media were available: audio, text and graphics. In a multi-user AR system, where a user interacts with the real-world, a much greater range of media types exist. Because each type of media has different physical properties, it has an impact on the importance of an object. One example of a medium is wireless communications. For two users to exchange data, they must be within the transmission range of their systems. Another example is that some tasks require a user to physically manipulate the object. Because the user has to be able to touch the object, the range over which the interaction can occur is highly limited.

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, has a finite transmission range. 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 propose to 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, in a firefighting scenario, such criteria might include whether an object is flammable or whether a street is wide enough to allow emergency vehicles to gain access. In general, the relevance is not binary-valued, but is a continuum that is normalized to the range from 0 (irrelevant) to 1 (highly relevant). For example, for the flammability criterion, the relevance might indicate the object's combustability.

Because the list of criteria and the measure of each object with those criteria is highly domain dependent, we assume that the choice of the criteria and the scoring of each object with respect to that criteria is carried out by one or more *domain experts*. In general, defining the list of criteria and evaluating objects with respect to those criteria is likely to be extremely difficult. However, such expertise is available in some applications domains. For example, the sniper avoidance system described in the next section relies on US Army Training manuals that precisely codify building features and configurations [2].

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.

We now formalize the framework.

4.2 The Filtering Framework

The state of the jth user is $\mathbf{u}_j$, where

$$\mathbf{u}_j = \begin{bmatrix} \mathbf{p}_j \\ \mathbf{t}_j \end{bmatrix}. \tag{1}$$

The user's objective state is $\mathbf{p}_j$ and the user's task vector is $\mathbf{t}_j$. The task vector can be linked to ancillary task-specific information.

The user's focus, $\mathbf{f}_j^m$, is a function of the user's state and medium m and is given by the equation

$$\mathbf{f}_j^m = \mathbf{f}(\mathbf{u}_j, m). \tag{2}$$

The state of an object is only fully defined with respect to a particular user. Specifically, the state of the ith object

with respect to the jth user is $\mathbf{x}_i^j$ where

$$\mathbf{x}_i^j = \begin{bmatrix} \mathbf{o}_i \\ \mathbf{s}_i^j \end{bmatrix}. \quad (3)$$

The vector of objective properties is $\mathbf{o}_i$ and the vector of subjective properties is $\mathbf{s}_i^j$. The subjective properties are derived from the user's state and the object's objective state by a domain expert. This relationship is captured in the equation

$$\mathbf{s}_i^j = \mathbf{s}(\mathbf{o}_i, \mathbf{u}_j) \quad (4)$$

where $\mathbf{s}\cdot,\cdot$ is a function which represents the domain expert's analysis of the objective and subjective properties.

The nimbus of the ith object for the jth user in medium m is $\mathbf{n}_i^{j,m}$,

$$\mathbf{n}_i^{j,m} = \mathbf{n}(\mathbf{x}_i, \mathbf{u}_j, m). \quad (5)$$

Once the focus and the nimbus regions have been calculated, the level of interaction which occurs between a given focus and nimbus, $q_i^{j,m}$, must be calculated. This is given by the equation

$$q_i^{j,m} = \mathbf{l}(\mathbf{f}_j^m, \mathbf{n}_i^{j,m}) \quad (6)$$

where $\mathbf{l}(\cdot,\cdot)$ is the level of interaction function.

5 A Sniper Avoidance System

In this section, we describe an implementation of the filtering algorithm for a sniper avoidance application. In many law enforcement, hostage rescue and peace keeping missions the threat of snipers cannot be ignored. Snipers, armed with powerful and accurate weapons, exploit the 3D nature of the urban environment. The system described here is intended to help counter sniper threats in two ways. First, the system provides safe routing through a urban environment avoiding sniper threats. Second, it provides information that is relevant for planning an operation to disarm a sniper.

Task analysis from the US Army Training Manual on urban operations [2] suggests that six tasks are relevant:

1. **Route:** The user needs to travel from point A to point B. The system should show the positions of the snipers and the user's destination point at all times. The system should also show fire hazards (an area which is well-suited for a sniper to set up an ambush) at a medium range from the user's position. The system should also provide general orientation information such as the names of streets and nearby buildings.

2. **Building Entry:** In this task, a user enters a building where a sniper is believed to be located. The system should show the suspected sniper positions, possible ambush sites, and the windows and doors around in the building where the sniper is located.

3. **Strategical Planning:** For this task the user must develop a broad plan of action which might cover many city blocks. The system should provide the user with a broad view of the environment which does not necessarily convey much depth. For example, the user will want to see buildings but does not need fine-grained data such as the location of individual windows or doors.

4. **Tactical Planning:** The user must develop a detailed plan of action involving a small area (typically only a few buildings). This task is invoked when a sniper position has been reported and the user is developing plans to negotiate around the target area. Fine-grained data such as the positions of windows might be used by the sniper is highly relevant.

5. **Offensive:** The user is going to perform offensive maneuvers to disarm the sniper. The system should show information about the known snipers only.

6. **Defensive:** The user is going to perform defensive maneuvers as the result of a sniper attack. The position of the sniper, buildings that offer good shelter and fire hazards should all be shown.

Two types of media can be defined. They are:

1. **Visibility:** Are two objects visible?

2. **Offensive Capabilities:** Are the users within range of a sniper's weapon?

An initial implementation of the filtering algorithm has been completed for a single user in a mobile environment. This implementation defines the user state $\mathbf{u}_j$ (objective plus task vector), the object state $\mathbf{x}_i$ (objective state and importance vector) and provides sample implementations of Equations 2 to 6. Because this is a prototype, many of the parameters have not been fully defined from empirical studies and so the burden of setting the parameters has been placed directly on the user. To minimize this burden we followed the approach described in [1] and provided the user with a set of interactive controls. These interactive controls consist of a set of sliders which are visible in the user's head mounted display and can be controlled by a trackpad. Furthermore, we only consider the problem of working in the medium of offensive capabilities.

5.1 User State

Objective properties:

- **Location:** The position and orientation of the user. This is measured directly by the user's tracking system.

Component	Meaning
1	Route
2	Building entry
3	Strategic planning
4	Tactical planning
5	Offensive
6	Defensive

Table 1. The Structure of the Task Vector

Subjective properties:

- **Task vector:** This consists of the current user's goal. The components are defined in Table 1.
- **Focus:** This is represented as a square centered on the user's current location.

5.2 Object State

Objective properties:

- **Type:** An enumerated quantity that specifies the object type. This can be qualified by subtype information that provides more data. For example, a `DOOR` object can be designated as a `MAIN_ENTRANCE`.
- **Location:** The position and orientation of the entity. Because the objects are stored in a hierarchy, an entity's location is specified with respect to the location of its parent.
- **Size:** A bounding box. For the i object the lengths of the sides are given by the triple of numbers $[v_0^i, v_1^i, v_2^i]$.
- **Impact zone:** This is a range over which the object has direct impact on its environment. This is current modeled as an axially-aligned box whose sides are of length $[z_0^i, z_1^i, z_2^i]$.

Subjective properties:

- **Importance vector:** The importance vector is six-dimensional. The criteria are listed in Table 2.
- **Nimbus:** The calculated nimbus of the object. The algorithm for calculating the nimbus is described below.

5.3 Importance Vector Calculation

The importance vector is calculated for each object by domain experts according to Equation 4. With this prototype system, only the following two factors are utilized:

- Taller buildings tend to confer a greater tactical advantage because, a user at the top of the building, has a larger field of view. This tends to increase the weight on criteria 2 (where sniper could prepare an ambush).
- The type of object is used in the strategical and tactical planning tasks. Specifically, if the user is in strategical planning mode and the object is a fine-grained building feature (window, door, etc.) its weight is reduced.

5.4 Focus Calculation

The focus is calculated from Equation 2. However, as explained, the focus is currently determined manually. It is a square bounding box whose dimensions can be continuously varied from between 0m and 500m.

5.5 Nimbus Determination

The nimbus is calculated using the following implementation of Equation 5. The nimbus $\mathbf{n}_i^{j,m}$ is a bounding box. The length of the dth side of this box is $n_{i,d}^{j,m}$ and its value is given by

$$n_{i,d}^{j,m} = (v_d^j + z_d^j) * (1.0 + \varsigma_i^{j,m}). \tag{7}$$

The terms v_d^j and z_d^j are the length of the size of the object and its impact zone in the d dimension. The final term $\varsigma_i^{j,m}$ is a user-task-object inflation term. To calculate its value, we score the importance vector against the user's task vector. To achieve this scoring, the task vector is first projected into importance vector space. We model this projection as the linear operation given by the matrix $\mathbf{M}$, where

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{8}$$

The element in the ith row of the jth column encodes whether the ith element of the object's importance vector is important for the jth task. For example, the fourth row encodes the fact that the system should show the user the possible location of civilians (the fourth element in the importance vector) only in the strategic and tactical planning modes (elements 3 and 4 in the task vector).

The scoring is then given by

$$\varsigma_i^{j,m} = 0.2 \min\left(1, (\mathbf{M}\mathbf{t}_j)^T \mathbf{s}_i^j\right). \tag{9}$$

Component	Meaning
1	Known sniper position
2	Where sniper could prepare an ambush
3	Shelter from sniper attacks
4	Where civilians may be gathering
5	Where friendly forces are located
6	Way point on a route which a user is following
7	Manually highlighted by user

Table 2. The Structure of the Importance Vector

```
Calculate nimbuses for all objects with initial object
  state and user goals.
loop forever {
      if (state of an object has changed) {
            Update nimbus of that object
            Filter that object
      }
      if (user's goal has changed) {
            Update nimbuses of all objects
            Filter all objects
      }
      if (user position has changed more than
         a threshold distance)
            Filter all objects
}
```

Figure 2. The pseudo-code of the main filtering loop.

5.6 Focus-Nimbus Interaction Calculation

The focus-nimbus interaction level, $q_i^{j,m}$, is calculated from the function $\mathbf{l}(\mathbf{f}_j^m, \mathbf{n}_i^{j,m})$ in Equation 6. We use the following approach. If the focus and nimbus boxes do not overlap then $q_i^{j,m} = 0$. If the user's position lies inside the nimbus then $q_i^{j,m} = 1$. However, if the focus and nimbus overlap but the user's position does not lie inside the nimbus, $q_i^{j,m}$ is 1.0 minus the minimum distance between the perimeter of the nimbus and the user's current location, divided by the length of the user's focus.

5.7 Filtering Architecture

In Figure 2 we provide the pseudo-code for the main filtering loop. This algorithm is completely dynamic—it can respond to any changes to the user or to the entities in the environment.

5.8 User Interaction and Display Cues

The filtering mechanisms described in the last subsections provide a means for automatically calculating the importance of buildings. However, the question remains as to *how* this information should be displayed to the user. A simple binary show/no show scheme has the problem that it is possible for configurations to arise within which a small change in the user's position can lead to an extremely large and disorientating change in the state of the graphical display. Therefore, based on our previous experience building situated AR displays [10], we use the following set of simple display cues to simplify the transition between different information display types. These cues are as follows:

1. Objects are faded in and out rather than appearing and disappearing. The alpha value for the ith object, α_i^j, is calculated by the ramping function

$$\alpha_i^j = \begin{cases} l_i^j/\beta & \text{for } q_i^{j,m} < \beta \\ 1 & \text{elswhere} \end{cases} \quad (10)$$

   where $\beta = 0.3$.

2. The user has the capability to select an object and define its importance. One element of the importance vector is "has the user selected this object?". This provides, in effect, a means by which the user can select an object and define the importance of that object.

5.9 Results

The effect of the filtering algorithm are demonstrated in Figure 3 which shows a pair of images captures from the actual output of the AR display mounted in a mannequin head that wears the display. The results show the effect of the system when it is running in the **Tactical planning** mode. In this mode, a user sees detailed environmental information. The image on the left shows the output from the system when filtering is disabled. As can be seen the image is highly cluttered — the system is showing the users data about the infrastructure of buildings behind the currently visible building. The effects of filtering are shown

in the right hand picture. As can be seen, the clutter has been eliminated. However, the system has not used a simple fixed-distance clipping strategy. This is illustrated by the fact that a reported sniper location on a building behind the visible building is visible.

A better example of the impact of the filter can be seen in Figure 4, which shows three frames grabbed from the frame buffer of the mobile graphics system. These three frames illustrate the effect of different task vectors on the user's output. Figure 4(a) shows the effect when filtering is disabled. The user is looking towards a building in the foreground. A sniper is visible in the background, but is hard to see due to the large amount of information which is being displayed. Figure 4(b) shows the view when the user switches to **Tactical Planning** mode. The system only shows the location of the sniper and the building that the sniper is believed to be resident in. Because the building is at some distance from the user, it is drawn more faintly to denote its lower importance. However, the persistent importance of the sniper means that it is drawn brightly. Figure 4(c) shows the same view but when the **Route** mode is enabled. In this view, the system shows the locations of threats (the sniper) and local landmarks (the foreground building).

Although we have yet to perform formal user evaluation studies, user response to the filtering algorithm has been extremely positive. Users have commented that the algorithm eliminates superfluous information and maintains critical data which is critical to a sniper avoidance system.

In the example shown, the system sustains 20 frames per second (stereo). Profiling reveals that the filtering algorithm, implemented in Java on the mobile computer (a 266MHz Pentium Pro), completely filters an environment of 150 objects in less than one millisecond. This performance is sufficient for our current development system.

6 Conclusions and Future Work

This paper has described an automated information filtering algorithm that we use to declutter the display of an experimental mobile AR system. The algorithm is based on the spatial model of interaction and utilizes a focus and a nimbus. We described a method for calculating the focus and nimbus which decomposes objects into objective and subjective properties. We demonstrated the use of this approach in a sniper avoidance system.

There are several areas of further work to be carried out:

- User studies and detailed domain analysis need to be carried out to refine domain expertise. This will be used to enhance the structure of the information vector and the evaluation of objects with respect to the appropriate criteria.
- The complexity of the environment model and of the criteria used to develop the focus and nimbus regions will be greatly enhanced. The current implementation only uses simple geometric descriptions (axially aligned bounding boxes) to model the environment and simple queries (box intersections) are used. We propose to extend our algorithm to incorporate line-of-sight and visibility constraints [9] and to use more sophisticated intersection algorithms [16].
- Future research algorithm can be combined with dynamic and flexible view management capabilities. Through the use of mechanisms such as constraint-based layout control, new annotations can be combined. The filter could be extended to provide priorities for the types of information which are being displayed.

Acknowledgments

This work is funded by the Office of Naval Research, Arlington, Virginia.

References

[1] C. Ahlberg, C. Williamson and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of ACM CHI '92*, pages 619–626, 1992.

[2] U. S. Army. Infantryman's guide to combat in built-up areas. Technical Report Field Manual No 90-10-1, U. S. Army, Washington D. C., May 1993.

[3] Y. Baillot, E. Gagas, T. Höllerer, S. Julier and S. Feiner. Wearable 3D Graphics for Augmented Reality: A Case Study of Two Experimental Backpack Computers. Submitted, 2000.

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

[5] S. Benford, J. Bowers, L. Fahlén, C. Greenhalgh, J. Mariani and T. Rodden. Networked Virtual Reality and Cooperative Work. *Presence*, 4(4):364–386, Fall 1995.

[6] R. Carey and G. Bell. *The Annotated VRML 2.0 Reference Manual*. Addison-Wesley, Reading, MA, 1997.

[7] T. P. Caudell and D. W. Mizell. Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. In *Proceedings of 1992 IEEE Hawaii International Conference on Systems Sciences*. IEEE Press, January 1992.

[8] K. Cheverst, N. Davies, K. Mitchell and G. S. Blair. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In *Proceedings of CHI' 00*, Netherlands, 2000.

(a) Original unfiltered view.

(b) Filtered version.

Figure 3. The effect of the information filter. The left image shows an unfiltered, cluttered view of the environment. The right image shows the same view when the filter is activated with a task of Tactical planning.

[9] S. Feiner, B. MacIntyre and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, July 1993.

[10] S. Feiner, B. MacIntyre, T. Höllerer and T. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *Proceedings of the International Symposium on Wearable Computers, Boston MA*, October 1997.

[11] H. Fuchs, M. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. Crawford, P. Rademacher, S. Drake and A. Meyer. Augmented Reality Visualization for Laparoscopic Surgery. In *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, October 1998.

[12] G. W. Furnas. Generalized Fisheye Views. *SIGCHI Bulletin: Proc. ACM Conf. Human Factors in Computer Systems, CHI*, 36(7):16–23, April 1986.

[13] W. A. Hoff, T. Lyon and K. Nguyen. Computer Vision-Based Registration Techniques for Augmented Reality. In *The Proceedings of Intelligent Robots and Control Systems XV, Intelligent Control Systems and Advanced Manufacturing*, volume 2904, pages 538–548, November 1996.

[14] T. Höllerer, S., Feiner, T. Terauchi, G. Rashid and D. Hallaway. Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers and Graphics*, 23(6):779–785, December 1999.

[15] R. Hull, P. Neaves and J. Bedford-Roberts. Towards Situated Computing. In *Proc. ISWC '97 (First Int. Symp. on Wearable Computers)*, pages 146–153, Cambridge, MA, October 13–14 1997.

[16] T. L. Kay and J. T. Kajiya. Raytracing complex scenes. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):269–278, 1986.

[17] B. MacIntyre and S. Feiner. Future Multimedia User Interfaces. *Multimedia Systems*, 4(5):250–268, 1996.

[18] T. Mori, K. Koiso and K. Tanaka. Spatial Data Presentation by LOD Control Based on Distance, Orientation and Differentiation. In *Proc. UM3 '99 (Int. Workshop on Urban 3D/Multimedia Mapping)*, pages 49–56, Tokyo, Japan, 1999.

[19] B. Shneiderman. *Designing the User Interface*. Addison-Wesley, third edition, 1998.

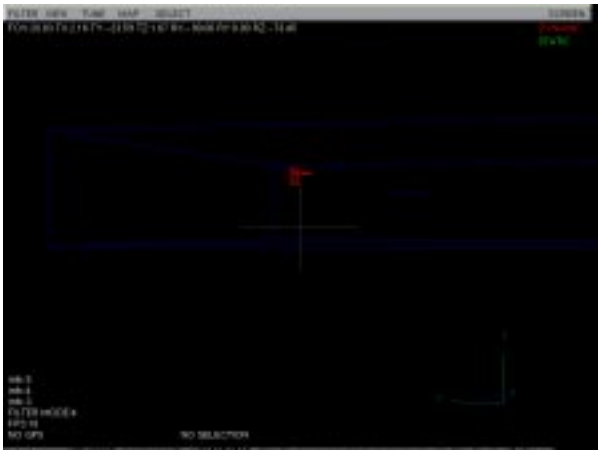
[20] H. Sowizral, K. Rushforth and M. Deering. *The Java 3D API Specification*. Addison-Wesley, 1997.

[21] A. W. Stedmon, R. S. Kalawsky, K. Hill and C. A. Cook. Old Theories, New Technologies: Cumulative Clutter Effects Using Augmented Reality. In *IEE International Conference on Information Visualisation '99, London, U.K.*, July 1999.

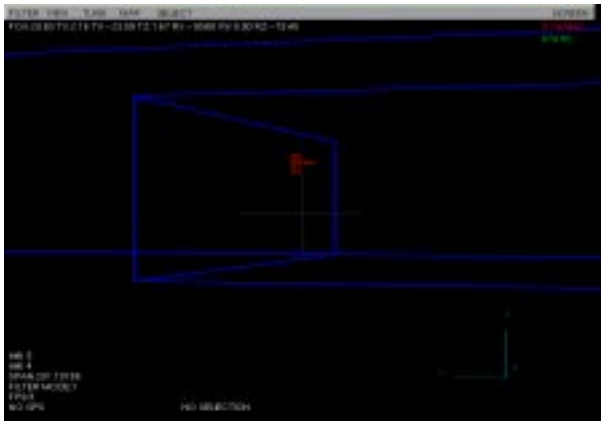
[22] A. van Dam, G. Stabler and R. Harrington. Intelligent satellites for interactive graphics. *Proceedings of the IEEE*, 62(4):483–492, April 1974.

[23] A. Webster, S. Feiner, B. MacIntyre, W. Massie and T. Krueger. Augmented reality in architectural construction, inspection and renovation. In *Proc. ASCE Third Congress on Computing in Civil Engineering*, pages 913–919, Anaheim, CA, 1996.

(a) Original unfiltered view.



(b) Filtering with **Tactical Planning** mode enabled.



(c) Filtering with **Route** model enabled.

Figure 4. Several images from the backpack computer captured directly from the system's frame buffer showing the effect of several different filter modes. It should be noted that the user's position is different from in Figure 3.