An Augmented Virtual Reality Interface for Assistive Monitoring of Smart Spaces

Shichao Ou, Deepak R. Karuppiah, Andrew H. Fagg, Edward Riseman, Roderic Grupen
Department of Computer Science,
University of Massachusetts,
Amherst, MA - 01003, USA
chao@cs.umass.edu

September 4, 2003

Abstract

Large sensor networks in applications such as surveillance and virtual classrooms, have to deal with the explosion of sensor information. Coherent presentation of data coming from such large sets of sensors becomes a problem. Thus there is a need to summarize events while retaining the spatial relationship between sensors. Also, such systems are prone to routine failures influenced by hardware, software, or the environment. To recover from such failures, fault containment can be achieved by using redundant sensors. In this paper, we define Fault Containment Unit (FCU), which has built-in alternative actions in the event of failures. However, the combinatorial explosion of alternatives in large scale sensor networks dictates that the design of FCU is hard.

Our strategy is to provide an augmented virtual reality interface to a human user by projecting the current state of the system, including camera orientation and objects being tracked, onto a virtual 3D world. We present an interface that : (i) offers different levels of detail when presenting information to user, (ii) allows the user to maintain a good spatial sense during sensor transitions, (iii) enables the user to dynamically assemble Fault Containment Units in response to emergencies, (iv) adapts to the current bandwidth availability, (v) provides mobility to the user, and (vi) allows shared interaction among users by immersing them in the same virtual workspace. Finally, we demonstrate three scenarios highlighting the above mentioned features.

1 Introduction

With the availability of cost effective sensors and processors, distributed sensor systems with hundreds of sensors are now becoming a reality. As a result, applications such as surveillance and virtual classrooms, now have to deal with

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
04 SEP 2003		-

4. TITLE AND SUBTITLE	
An Augmented Virtual Reality Interface for Assitive Monitoring of Smart Spaces	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Advanced Research Projects Agency,3701 North Fairfax Drive,Arlington,VA,22203-1714**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**17**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

the explosion of sensor information. Clearly, in the context of surveillance, the traditional *user interface* (UI) , such as a room of monitors each showing a live video stream from a corresponding camera, does not scale as the number of sensors grows. Switching between different camera streams on a single monitor is unavoidable when there are much fewer monitors than cameras. Thus the acquisition of "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and projection of their status in the near future" [4] , also called *situation awareness* becomes more difficult. More importantly, the engagement of user's spatial memory requires more effort, since the user would have to remember past events in forms such as "room 315" or "Camera 250", which do not explicitly convey any spatial relationships. Switching across large numbers of cameras can become extremely confusing while following an event of interest. These in turn drive up training costs and increase response time to emergencies. The importance of *situation awareness* becomes even more apparent when the sensors are steerable, i.e. cameras mounted on mobile robots or pan-tilt units. The user can quickly teleoperate such sensors only when he acquires a good spatial sense of the current viewpoint both before and after a sensor switch.

Traditional UIs rely on the user's own recognition ability and judgment to analyze the scene to determine what is interesting or suspicious. Human's cognition alone cannot be relied to detect suspicious activity from large numbers of cameras over extended periods of time and in extremely cluttered environments (Figure 1).

Figure 1: Complex cluttered environment shown from three cameras

Traditional monitoring systems are in general not adaptive since they have dedicated bandwidth requirements. Users cannot be mobile while interacting with such a system. For example, a night watchman walking around a building cannot instantly get access to what is happening around the corner or effectively share information with personnel in the control room.

Besides the problem of explosion of sensor information, large scale sensor networks are also prone to routine failures influenced by hardware, software, or the environment. To recover from such failures, fault containment can be achieved by using redundant sensors. We formalize this notion in section 3 by defining Fault Containment Unit (FCU), which has built-in alternative actions in the event of failures. However, the combinatorial explosion of alternatives in large scale sensor networks dictates that the design of FCU is hard.

A solution to the problem is to include human interaction in the decision making process. For instance, a human user can take an assistive role in a system with adjustable autonomy [16], when the system is incapable of achieving the task. Even when AI methods are applied, the system has to present the user with relevant information about the progress of the task being undertaken. At the same time, the data gathering process can be influenced by the user through interaction. In light of this, we believe that the design of an intuitive human-computer interface is crucial to achieving a harmonious working relationship between human and machine.

We have designed a system that can robustly track motion in an environment with many cameras located potentially far from each other. In the general case, the cameras could be both stationary and mobile, the environment could be cluttered, could be indoors as well as outdoors. By tracking motion from multiple cameras, a system-wide representation of the identity of people may be constructed and maintained in a consistent manner. The individual cameras perform motion segmentation and extract blobs that have significant motion. Using corresponding blobs from each camera, we can triangulate on the object and compute its 3D location and size.

In section 1.1, we review related work in the areas of user interfaces and tracking. In section 2, we introduce augmented virtual reality to overcome the shortcoming of traditional UI used in monitoring systems. In section 3, we describe a fault handling mechanism in systems with redundant resources and emphasize the difficulty in designing such mechanisms for large scale systems. In section 4, we present the experimental setup and show some scenarios where our system was applied. Finally, section 5, summarizes the work and gives directions for future work.

1.1 Related Work

A number of research efforts in aviation and military domains have shown that better understanding of terrains can be achieved by navigating through 3D interfaces [25, 14, 1]. Results from studies on spatial memory and user interfaces concur that “measures of spatial cognition strongly predict performance with computer interfaces”[3]. Cockburn and Mckenzie [3] have shown spatial arrangement of documents allows for rapid retrieval. The gaming industry has long since moved from 2D to 3D to provide a much richer and more immersive world that the players can freely roam around within. These provide evidence to the assertion that since we live in a 3D world, the most intuitive way to interact with remote spaces is through a 3D virtual environment where the user is able to explore the spatial configuration of the environment, engaging one’s natural abilities to interact with environment and construct internal cognitive maps of the space [8].

Virtual worlds have been used in many applications (especially in gaming) to allow multiple users, most likely from vastly different geographic locations, to work or play collaboratively in a shared immersive interactive space. In most gaming applications (Quake [15]) , virtual worlds do not resemble the real world. On the other extreme, projects that build a digital city such as Kyoto, Helsinki or Amsterdam [12] attempt to reconstruct a virtual city to match its real world counterpart down to fine detail, for example a convenience store, such that an effective immersive true-to life experience can be achieved. In the case of surveillance applications, although it is desirable to model the space as accurately as possible, we feel it is neither essential nor practical to do so.

Augmented Virtual Reality (AVR) is not a new concept. According to the taxonomy of mixed reality by Tamura and Yamamoto [22], AVR belongs to the definition of Class B (Video see-through) or Class C (On-line tele-presence) depending on whether the real-world imagery comes from scenes directly in front of the user or a remote site. Numerous augmented reality systems have been built to augment the real-world scene with virtual objects or text to provide information to the user [23, 13, 20, 2].

In recent years, multi-sensor networks have been designed to do human tracking and identification ([10], [18], [17], [19] , [21]). Trivedi et al.[24] have proposed an integrated system of active camera network for human tracking and recognition. Matsuyama et al.[7] present a practical distributed vision system based on dynamic memory. In our previous work [26], we have presented a panoramic virtual stereo for human tracking and localization in mobile robots. However, most of the current systems emphasize on vision algorithms, which are designed to function in a specific network. Karuppiah et al. [6] present a distributed control architecture in which run-time behavior is both pre-analyzed and recovered empirically to inform local scheduling agents that commit resources autonomously subject to process control specifications has been presented.

2 Augmented Virtual Reality Interface

2.1 Virtual 3D environment

We have discussed in detail the shortcomings of traditional live video stream in section 1. As opposed to the traditional approach, virtual environment is immersive, i.e. user can freely move about without abrupt spatial changes. The smooth transition using virtual fly-through (Figure 8(b)) enables us to synthesize those views that are not serviced by real-world cameras. This is very important for achieving *situation awareness* using the user's natural spatial cognition abilities, as shown in the many studies discussed in section 1.1. Information can now be stored and accessed spatially. For example, missed events can be stored at the correct location and can then be accessed later for analysis by utilizing the user's spatial memory, rather than the user having to remember a room number or camera ID. This reduces the cognitive load of the user.

With virtual environment interface, network bandwidth requirement can be greatly reduced. Only abstract information such as (x,y,z) coordinates, or the color of the tracked object, are sent across the network. These are much more light-weight representations than the raw video data stream. Moreover, we can talk about different level of detail when presenting information to the user, thus avoiding information overload. For example, when multiple subjects moving about in an area are being precisely tracked, the system does not need to display these avatars in the interface unless any of the subjects has moved into or close to a restricted area. Only then should the user be alerted to the locations of the tracked subject.

Virtual environment allows multiple remote users to work and share information in the same virtual workspace. For example, in a surveillance scenario, a security personnel can be virtually transported to a night watchman's location and work with him through the interaction in the virtual environment.

However, virtual environment interfaces are not without their disadvantages. To begin with, pre-construction of the virtual environment is required. Depending on the types of information needed to be conveyed to the user, virtual objects or avatars with different levels of detail have to be built. More importantly, events may be missed since VR interface rely on sensors to provide abstract information for display. For example, if someone is picking a lock, from the visual interface he may seem to be merely standing still beside a door. This is because there is no sensor to detect the lock-picking motion. Even if there was one, if we did not model the lock-picking motion before hand, this motion cannot be displayed.

The problems mentioned above do not appear in the traditional live video stream approach, since it does not extract nor throw out any information.

2.2 Augmenting Virtual Reality

In this section we introduce the concept of Augmented Virtual Reality (AVR) that enables the user to monitor the environment in both the abstract information space and the real space. Such a mix takes advantage of best of both interfaces. As described in section 1.1, this is not a new concept. In fact, our implementation falls into the category of class C - Online tele-presence [1]. However, we feel that our approach is unique because we propose to augment the virtual world with real video streams as opposed to the traditional augmented reality applications[13, 20] that overlay text or virtual avatars on top of video streams. More importantly, this process happens in real-time . The proposed AVR interface works in 3 modes:

- a pure virtual world mode that displays abstract information extracted by the sensors.

[1] merging video images transmitted from a remote site and virtual images, giving the observer a mixed view of two different different worlds

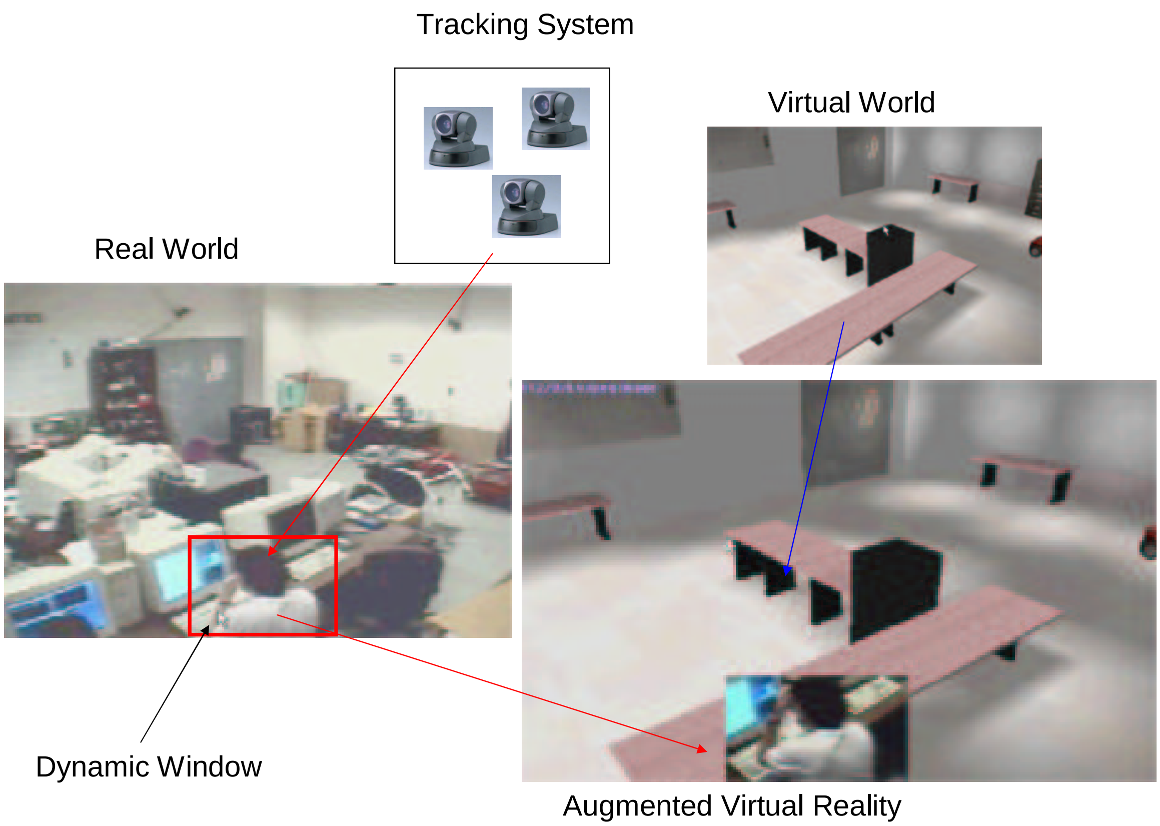

Figure 2: Augmenting virtual reality : Using information extracted by the tracking system, the AVR interface overlays the real-world imagery of the tracked object on top of the virtual world to both convey spatial sense and to save bandwidth. Note the virtual and real camera views are aligned such that the real-world image appears in the correct location within the virtual view.

- a full video stream mode that streams real videos from each camera.
- a mixed mode that overlays the most interesting portions of the real world on top of the virtual world such that the virtual world conveys context and spatial relationship of the scene, while the dynamic window displays the real-time imagery of scene (see figure 2).

The AVR interface augments the virtual world with real-life imagery to adapt to different bandwidth conditions, while at the same time provides the user with different levels of detail in presenting the information. For example, when high bandwidth is available, the user may choose to monitor the space in either the full-video stream mode, or the mixed mode to block out unwanted information, while the virtual fly-through between cameras provides immersiveness and helps to maintain spatial sense. When operating in mid or low bandwidth conditions or when the feature extraction in the sensors are completely reliable , the user may decide to monitor the scene using the pure virtual environment augmented with partial display real-world and only occasionally switch to the live video stream for verification purposes.

One of the disadvantages of pure VR is that, both the virtual environment and the avatars are required to be modeled in high-detail in order to accurately represent the real space. This is not required in AVR, since AVR will introduce some real-world details through either the mixed mode or the full video streaming mode.

3 Fault Containment Unit Hierarchy

In general, complex systems should be designed using redundant resources with the expectation that failures caused by some subset of resources can be overcome by others. To formalize this dynamic reallocation of resources while achieving a task objective, we define a Fault Containment Unit [6] as a fundamental way to specify tasks in our system. A containment unit is bound with a set of resources needed to accomplish the task with built-in modes to handle failures. In the extreme case where fault containment is not possible, a status report is communicated to the instantiating process of the containment unit. Thus a containment unit itself can be a resource to another containment unit with a higher level task specification. A hierarchy of containment units (Figure 4) is used in this work to perform various tasks in our the smart space such as the localization of people and robots, and the recognition of people. In our environment, faults can be generated by failure of hardware (sensors, robots, CPU, etc.), software (algorithms, controllers, etc.), communication, etc.

Below we show how containment units are used to manage resources in our system. Two low-level controllers that run on our pan-tilt-zoom (PTZ) cameras are the saccade controller that moves the camera towards the direction of an interesting feature e.g. motion, and the foveate controller that brings the feature of interest to the center of the field of view. Figure 3 shows a schema that can perform a saccade followed by foveate task. This schema also generates reports that describe its own behavior like *hardware fault, no target, target lost and heading report*. If a target feature is detected and foveated, then the sensor achieves state $X1$ where the feature is actively tracked. As long as the actions of the foveation controller preserves this state, a heading to the feature is reported. In all other cases, an appropriate report describing the nature of failure is generated.

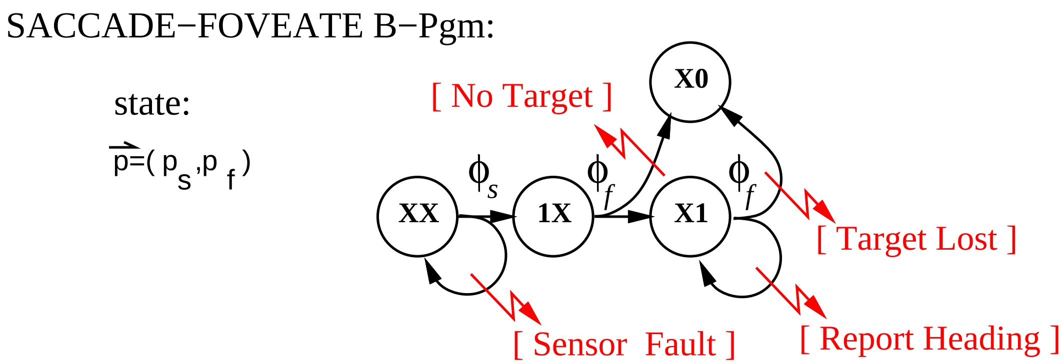

Figure 3: Containment unit wrapper for the saccade-foveate model

When two instances of the SACCADE-FOVEATE FCUs are simultaneously in state $X1$ and they are driven by features derived from the same subject, then there is sufficient information for triangulating the subject. A higher level containment unit called LOCALIZE FCU receives the event streams generated by two subordinate SACCADE-FOVEATE FCUs under its management and produces a report regarding the location of the subject. The subject of interest may at times be moving or stationary. In the former case, the LOCALIZE FCU may have to actively manage the subordinate FCUs, while in the latter case it can instantiate an MONITOR FCU that continuously confirms the presence of the stationary feature in the last known location.

At the highest level a FCU supervisor may instantiate multiple LOCALIZE FCUs each of which are responsible for maintaining a robust track of single subject of interest. Over time, the event streams coming from lower levels are used to build and update a collection of features that describe each subject. When a LOCALIZE FCU reports a lost track, the annotation of features to the corresponding subject are handed off to a new instantiation of LOCALIZE FCU with a different set of resources that are best placed to take over the tracking.

3.1 User as the top level in Containment Unit hierarchy

When the number of resources available to a containment unit is large, there is an exponential explosion in the choices for resource allocation, and offline hand-coding or prioritizing different courses of actions is quite difficult. Alternatively, user interaction at the highest level of the hierarchy can be effective as humans can react to situations using prior experience and dynamically reconfigure resources in a FCU and thereby recover the system from fault state. This removes the need to presuppose all but the most routinely anticipated failure modes.

The proposed user interface provides a direct means to interact with the Fault Containment Unit hierarchy as shown in figure 4. At the lowest level, each sensor is assigned to a TRACK FCU. TRACK FCU reports valid moving objects in the sensor's field of view. When tracking is unsuccessful, TRACK FCU generates fault reports and report to the higher level. LOCALIZE FCU receives event streams from subordinate TRACK FCUs to localize moving objects by triangulation. FCU Supervisor manages its subordinate FCUs by monitoring their fault states. At the highest level, the user interacts with the FCU hierarchy through AVR UI, and can take actions such as monitoring the state of LOCALIZE FCU or instantiating a new TELEOPERATE FCU.

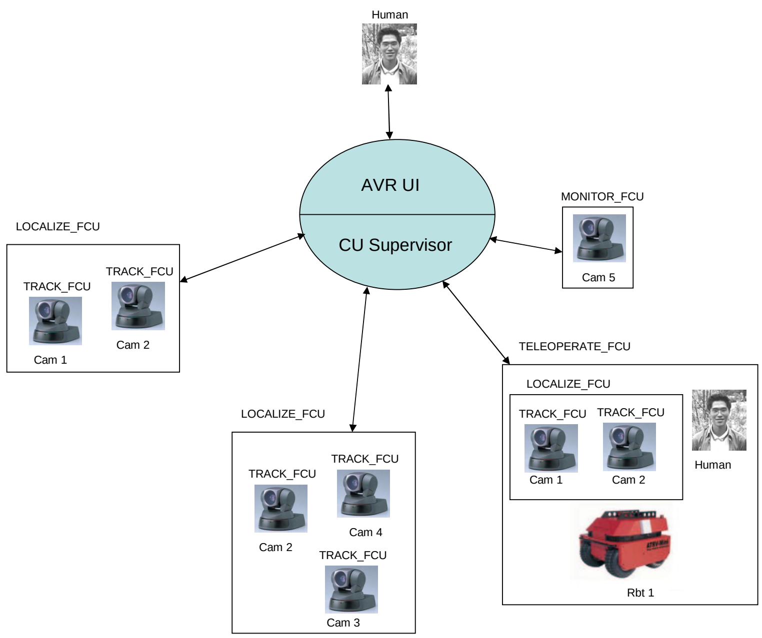

Figure 4: Fault Containment Unit Hierarchy

4 Experimental Setup

The UMass Smart Space has five Sony PTZ EVI-D100 cameras mounted on the walls and an ATRVJr mobile robot equipped with a fixed camera. Our compute rack consists of a cluster of six VMIC single board computers each with a 928 Mhz processor and 256 MB RAM. The nodes in the cluster share a 100Mbps and a 1000Mbps ethernet link and a wireless access point to communicate with the robot. Each node has a Leutron vision frame grabber to which a camera is connected. Using NDDS real-time publish-subscribe middleware [11], each node acts as a server of video and track information extracted from the camera. To create the virtual version of the smart space, room dimension measurements were taken, and likewise the position of the cameras. Prominent objects such as tables were placed in virtual space roughly in alignment with the placement of the real objects to function as references. Other details such as computers on desks, chairs were not modeled (and not necessary) for reasons discussed in 2.2. The robot control interface was implemented using Player/Stage [9]. The AVR interface was implemented using Genesis3D [5]. The overall system architecture is shown in figure 5

At startup, the AVR interface renders the virtual world, and uses NDDS middleware to create subscriptions of relevant information from the smart space. The camera models in the AVR interface are thus synchronized with their real-world counterparts. This feature enables smooth transitions between real and virtual views owing to the identical perspective. When the user requests for full video mode from any camera in the AVR interface, a subscription is activated to the corresponding video stream. Moving object locations published by the FCU supervisors are rendered in the AVR user interface as avatars. Using the object locations, mixed mode is presented to the user upon request. The user also can teleoperate robots in the smart space using this interface. Each robot's state is updated using both its published odometry as well as its track information from the FCU supervisor.

We present three real-life scenarios in which our system was tested to highlight the efficacy of our interface in those situations. The first scenario demonstrates robust tracking maintained under user supervision. Figure 6 shows the top-down view of our smart space room. Two containment units FCU1 and FCU2 were instantiated with two cameras each. The green(light shade) and brown(dark shade) overlays indicate the valid coverage area for FCU1 and FCU2 respectively. The FCU hierarchy automatically switches between the fault containment units to track a moving subject (red trail) and presents this information to the user. The user performs the supervisory role by ensuring that the instantiated FCUs are adequate for the task.

Figure 7 shows an extremely cluttered environment with multiple moving objects. The AVR interface in the mixed mode shows the user interesting objects (ranked based on their motion history) in the scene through a dynamic window around the real tracked object in real-time. We believe such interface offers an extra mode of information presentation to the user, which reduces uninteresting clutter in a scene and therefore reduces user's cognitive load.

The last scenario shown in Figure 8 demonstrates the dynamic reconfiguration of FCU with user intervention in the FCU hierarchy. Initially, the smart space tracks a moving person, who later tries to avoid the tracking system by hiding under a table, out of the view of all the cameras. When the system loses track of the person, it notifies the user since it is unable to recover from this fault by itself. Playing the assitive role, the user reacts by teleoperating the cameras, switching between different streaming modes and tries to recover the lost object, but is unable to do so. The important

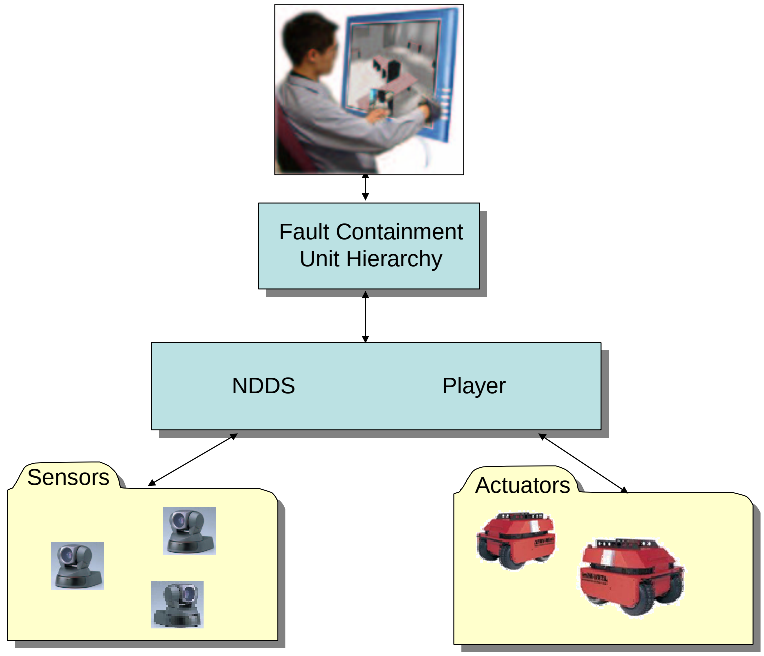

Figure 5: System Architecture

thing to note here is that in spite of continuous switching between cameras, due to the smooth transitions, the user is able to maintain a good spatial sense. After arriving the camera, the user switches to the full video mode and attempts to locate the missing person. Not finding the lost person, the user teleoperates a mobile robot to explore the vicinity of the last tracked location using a mixture of tele-presence in the robot and different wall-mounted cameras. Finally, he uncovers the person hiding under the desk and thus recovers the system from the fault state and the system resumes tracking him. This demonstrates the achievement of successful tracking failure containment through the use of AVR interface, and the effectiveness of placing the user in the loop.

The videos for the above scenarios can be accessed at
http://128.119.244.148/Research/Distributed_Control/PerComm04/

5 Conclusions and Future Work

This paper presents an implementation of a fault-tolerant augmented virtual reality interactive monitoring system. Each camera in the system tracks moving objects in its field of view, and the triangulated 3D location of objects are sent to a virtual reality interface which presents this information to the user in the form of virtual avatars. The virtual interface is augmented by partial or full real video streams on a need basis resulting in bandwidth flexibility. We believe that our

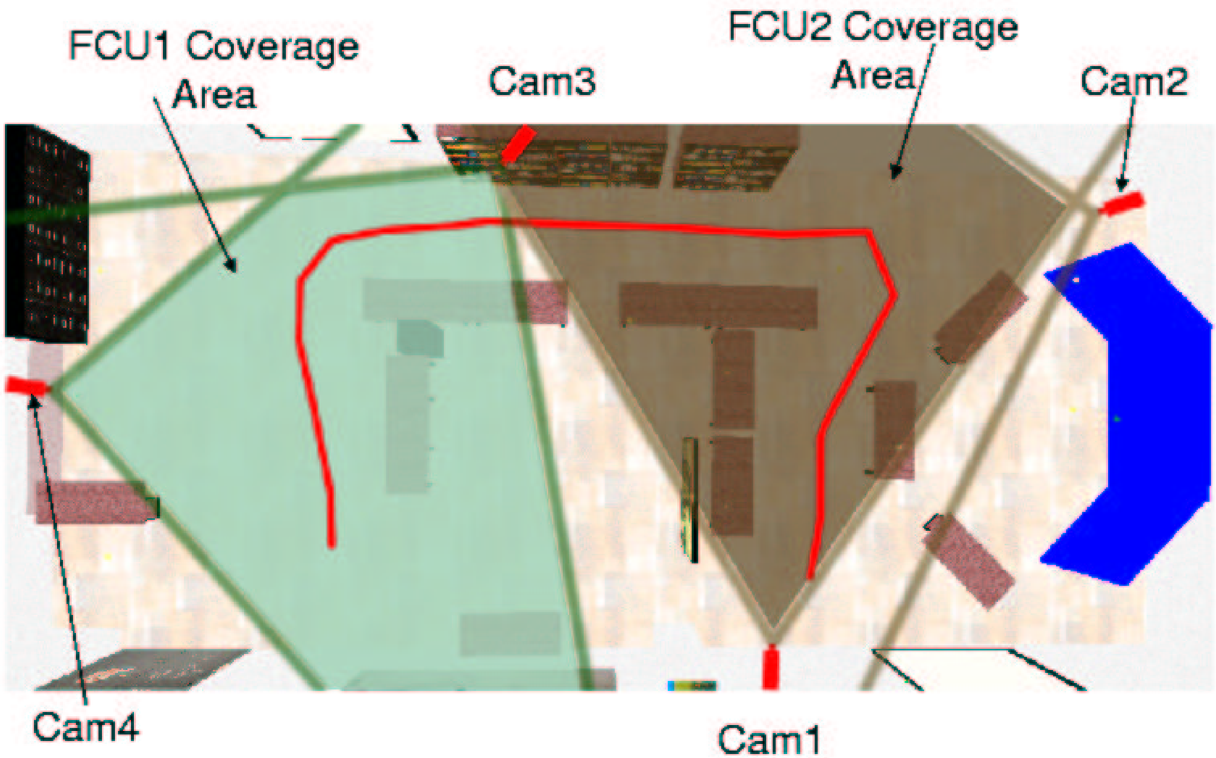

Figure 6: Robust tracking under user supervision. This is a top-down view of a room. Two containment units FCU1 and FCU2 are instantiated with two cameras each. The green(light shade) and brown(dark shade) overlays indicate the valid triangulation regions for FCU1 and FCU2 respectively. The FCU hierarchy automatically switches between the fault containment units to track a moving subject (red trail) and presents this information to the user. The user performs the assitive role by ensuring that the instantiated FCUs are adequate for the task.

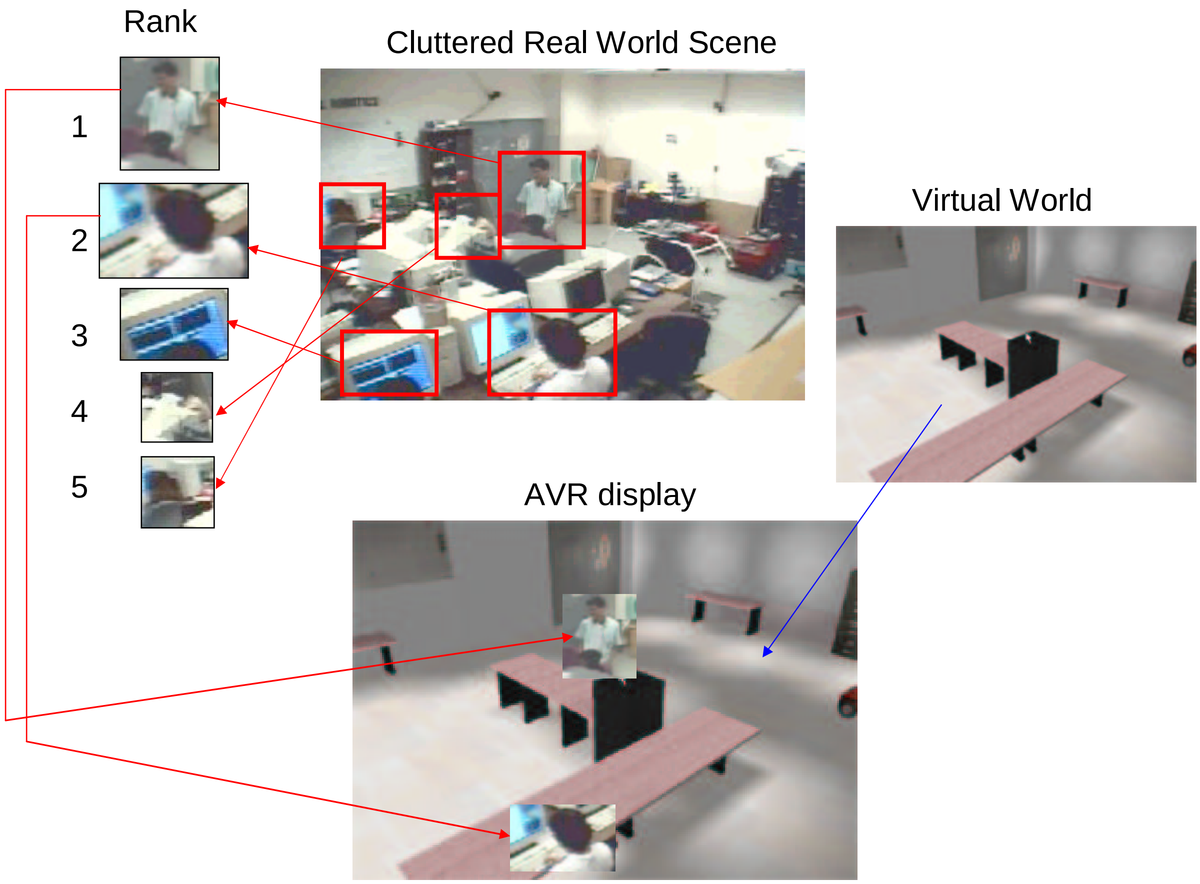

Figure 7: Attention focus through AVR. This figure shows an extremely cluttered environment with multiple moving objects. The AVR interface in the mixed mode shows the user the interesting objects (ranked based on their motion history) in the scene through a dynamic window around the real tracked object in real-time.

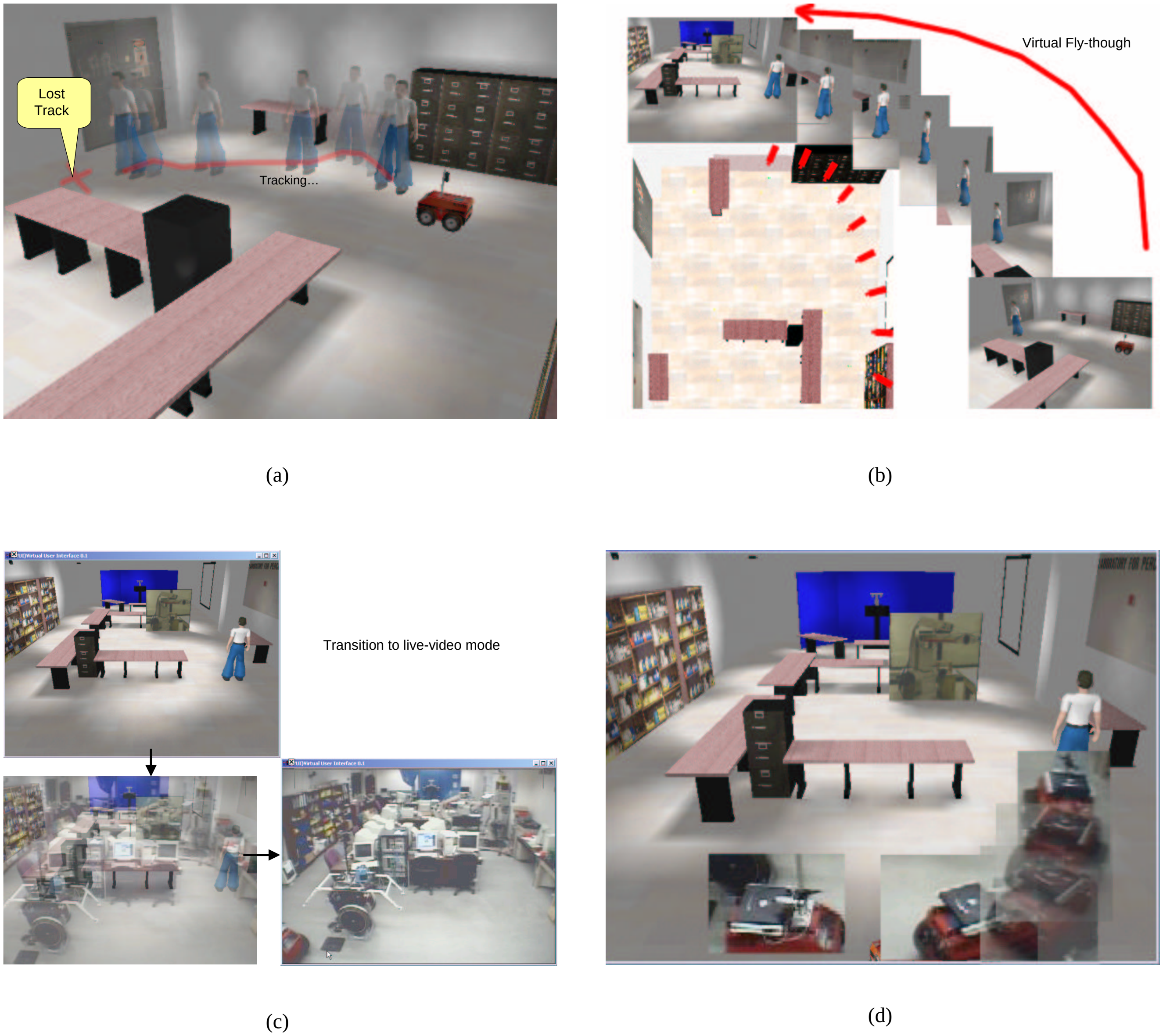

Figure 8: Dynamic reconfiguration of FCU with user intervention in the FCU hierarchy : (a) The smart room tracks a moving person, who later tries to avoid the tracking system by ducking down under a table, out of the view of all the cameras. When the system loses track of the person, it notifies the user since it is unable to recover from this fault by itself; (b) Playing the assitive role, the user reacts by teleoperating the cameras, switching between different streaming modes and tries to recover the lost object, but is unable to do so. The important thing to note here is that in spite of continuous switching between cameras, due to the smooth transitions, the user is able to maintain a good spatial sense; (c) This figure shows the transition from the pure virtual mode to full video mode and attempt to locate the missing person; and (d) Not finding the lost person, the users teleoperates a mobile robot to explore the vicinity of the last tracked location using a mixture of tele-presence in the robot and different wall-mounted cameras.

interface effectively engages the user's spatial memory, allowing the user to quickly acquire situation awareness.

The use of mobile robot as an actuator in the Fault Containment Unit to regain tracking of the target demonstrates the utility of having the user manage the hierarchy of containment units with large number of resources.

5.1 Future work

User interaction provides valuable dynamic control information for efficient reactions to urgent unanticipated situations that could be learned by the system, allowing interaction in similar contexts in future to be minimized.

As an extension to the attention focus scenario (Figure 7), we can imagine more sophisticated methods for the selection of the most interesting object(s). Using supervised learning approach, the user can teach the system to select an *interesting* object(s) selection policy that would balance between guaranteeing high probability of presenting suspicious activities in the scene while not causing information overload that would fatigue him quickly. Further studies need to be carried out to evaluate the interface for different parameters like reduction of user fatigue using spatial memory or time to acquire situation awareness.

Acknowledgments

The authors gratefully acknowledge support for this work from DARPA/ITO DABT63-99-1-0004(MARS), NSF CDA 9703217(Infrastructure) and NASA NAG9-1445#1.

References

[1] W. Barfield. and C. Rosenberg. Judgements of azimuth and elevation as a function of monoscopic and binocular depth cues using a perspective display. *Human Factors*, 37(1):173–181, 1995. 3

[2] M. Billinghurst and H. Kato. Collaborative agumented reality. *Communications of the ACM, July 2002, Vol. 45*, pages 64–70, 2002. 4

[3] A. Cockburn and B. McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of CHI*, 2002. citeseer.nj.nec.com/article/cockburn02evaluating.html. 3, 3

[4] M. R. Endsley. Towards a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995. 2

[5] Eclipse Entertainment. Genesis3d. http://www.genesis3d.com. 10

[6] D. R. Karuppiah et al. Software mode changes for continuous motion tracking. In P. Robertson, H. Shorbe, and R. Laddaga, editors, *Self-Adaptive Software*, volume 1936 of *Lecture Notes in Computer Science*, Oxford, UK, April 17-19 2000. Springer Verlag. 4, 7

[7] T. Matsuyama et al. Dynamic memory: Architecture for real time integration of visual perception, camera action, and network communication. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, Hilton Head Island, SC, June 2000. 4

[8] A. H. Fagg, S. Ou, T. R. Hedges, M. Brewer, M. Piantedosi, P. Amstutz, A. Hanson, Z. Zhu, and R. Grupen. Human-robot interaction through a distributed virtual environment. Workshop on Intelligent Human Augmentation and Virtual Environments, 2002. 3

[9] B. P. Gerkey and R. T. Vaughan. Most valuable player: A robot device server for distributed control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, pages 1226–1231, 2001. `http://playerstage.sourceforge.net`. 10

[10] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time system for detection and tracking people in 2.5d. In *Proc. of the 5th European Conf. on Computer Vision*, Freiburg, Germany, June 1998. 4

[11] Real-Time Innovations Inc. Ndds: The real-time publish-subscribe network. `http://www.rti.com/products/ndds/literature.html`. 10

[12] T. Ishida. Digital city kyoto. *Communications of the ACM, July 2002, Vol. 45*, 2002. 4

[13] T. Jebara, C. Eyster, T. Starner, and A. Pentland. Stochasticks : Augmenting the billiards experience with probabilisitic vision and wearable computers. In *First International Symposium on Wearable Computers ISWC'97*, 1997. 4, 5

[14] M. John, H. Oonk, and M. Cowen. Using two- dimensional and perspective views of terrain. Technical Report 1815, SPAWAR Systems Center, 2000. `www.nosc.mil/sti/publications/pubs/tr/1815`. 3

[15] J. E. Laird. It knows what you're going to do: adding anticipation to a quakebot. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 385–392, Montreal, Canada, 2001. ACM Press. 4

[16] D. Musliner and Cochairs B. Pell. *Agents with Adjustable Autonomy : Papers from 1999 AAAI Spring Symposium*. The AAAI Press, 1999. 3

[17] A. Nakazawa, H. Kato, and S. Inokuchi. Human tracking using distributed vision systems. In *In 14th International Conference on Pattern Recognition*, pages 593–596, Brisbane, Australia, 1998. 4

[18] Kim C. Ng, H. Ishiguro, Mohan M. Trivedi, and T. Sogo. Monitoring dynamically changing environments by ubiquitous vision system. In *IEEE Workshop on Visual Surveillance*, Fort Collins, Colorado, June 1999. 4

[19] A. Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, January 2000. 4

[20] W. Piekarski and B. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *Fifth International Symposium on Wearable Computers ISWC'01*, 2001. 4, 5

[21] T. Sogo, H. Ishiguro, and Mohan. M. Trivedi. *Panoramic Vision: Sensors, Theory and Applications*, chapter N-Ocular Stereo for Real-time Human Tracking. Springer Verlag, 2000. 4

[22] H. Tamura and H. Yamamoto. Vision and graphics in producing mixed reality worlds. In *IEEE and ATR Workshop on Computer Vision for Virtual Reality Based Human Communications*, pages 78 –85, Jan 1998. 4

[23] H. Tamura, H. Yamamoto, and A. Katayama. Mixed reality: future dreams seen at the border between real and virtual worlds. *Computer Graphics and Applications, IEEE*, 2001. 4

[24] M. M. Trivedi, K. Huang, and I. Mikic. Intelligent environments and active camera networks. *IEEE Transactions on Systems, Man and Cybernetics*, October 2000. 4

[25] C. Wickens, C. Liang, T. Prevett, and O. Olmos. Egocentric and exocentric displays for terminal area navigation. In *Human Factors and Ergonomics Society 38th Annual Meeting*, pages 16–20, 1995. 3

[26] Z. Zhu, K. Deepak Rajasekar, E. Riseman, and A. Hanson. Panoramic virtual stereo vision of cooperative mobile robots for localizing 3d moving objects. In *Proceedings of IEEE Workshop on Omnidirectional Vision - OMNIVIS'00*, pages 29–36, Hilton Head Island, SC, June 2000. 4