Visual Contextualization and Activity Monitoring for Networked Telepresence

Douglas A. Fidaleo, R. Evan Schumacher, and Mohan M. Trivedi
University of California San Diego
9500 Gilman Dr.
La Jolla, CA, USA
{dfidaleo|rschumac|mtrivedi}@ucsd.edu

ABSTRACT

The context of an environment is defined by a complex interrelationship between past, present, and future events and properties of the events' surroundings. While the present provides a current and immediate interpretation of our surroundings and enables immediate decisions, historical data enables post-mortem analysis of an incident as well as a foreshadowing of future events that can in turn affect the context upon which current decisions depend. Achieving context awareness therefore requires extraction, capture, and interpretation of multiple modes of information from different temporal and spatial contexts. Contextualization of this data involves embedding the information into a single familiar virtual environment to assist a human operator in making sense of the volumes of collected sensory data.

This paper details a prototype system we are developing to facilitate future context awareness research as well as support, integrate, and augment existing context focused research developed in our laboratory. The core modules of the system are described including the Networked Sensor Tapestry (NeST) architecture for sensor integration, processing, and context archiving and querying with a focus on the Context Visualization Environment (CoVE): a 3D environment for visual fusion of multimodal sensor data. Several real-world applications are presented that are built on top of these components.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Arificial, augmented, and virtual realities*

General Terms

Experimentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ETP'04, October 15, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-933-0/04/0010 ...$5.00.

Keywords

Context awareness, contextualization, visualization, surveillance

1. INTRODUCTION

1.1 Motivation

Crisis scenarios range in magnitude from an a earthquake in a densely populated city to a car accident at a local intersection. The current state of the art for incident mitigation involves passive reporting of the incident to a crisis management facility (such as the local police station), deployment of first responders for critical assistance and incident evaluation, and subsequent deployment of more focused backup resources.

There are several problems with the current crisis response scenarios. Critical time is lost before the event actually gets reported and when it does get reported the information conveyed may be incomplete, biased, or inaccurate due to the inexperience and/or emotional state of the observer. First responders are charged with the responsibility of managing the current crisis, as well as collecting information and relaying it to a control center to evaluate the potentially changing scenario. This can distract the responders from their primary task, and potentially lead to errors in judgement.

These problems can be solved or their impact lessened by reducing the burden on humans for detecting, reporting, and evaluating information surrounding the crisis event. Considerable research is necessary, however, before these systems can operate entirely automatically with a level of reliablity that surpasses current human-managed mitigation processes. The work presented in this paper is an experimental system for collection, archival, and visual fusion of multisensory data (temperature, video, GPS, RFID) that we have developed to facilitate future research in context extraction from an arbitrary environment: a critical component of crisis mitigation.

1.2 Context Awareness

Dey et. al. define context as "any information that can be used to characterize the situation of entities (i.e. whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identitiy and state of people, groups, and computational and physical objects." [3].

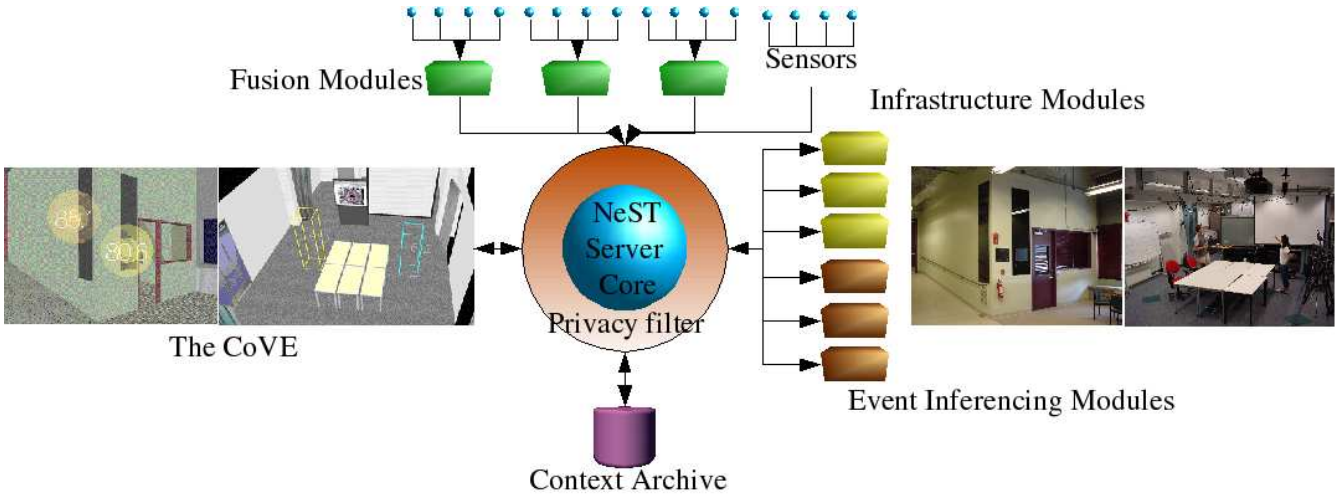

Figure 1: Components of the context awareness system including data collection hardware and software, data archiving and querying, context processing, and visualization and interaction.

By this definition, low-level sensor data collected from the event's environment are in fact context, however we distinguish this from high-level context by using the term *context data sources* to refer to information such as the location of a person, her identity, the interactions she has had with other individuals, and the temperature of the environment. These data sources can be analyzed when an event occurs (for example, the person enters a room) to answer higher-level context questions such as "Why did she enter?" which may involve additional mining of information by generating more context data such as her location before the event occured or a log of the last time she entered the same room. In this paper, we emphasize *contextualization* of the data sources: embedding the data in an environment that facilitates extraction of higher-level context.

Because of the critical nature of crisis events, for the forseeable future humans must be involved in context extraction processes. However, the collected sensory data can appear as hoardes of disjoint fragments of information that cannot be effectively utilized by an end user (for example, a police station dispatch office) in their raw form. An important contextualization step is therefore to focus and anchor the data into a familiar environment to provide higher level semantic awareness to the user.

The most intuitive anchor for sensory data is the actual environment from which the data is being collected. As such, video streams provide an effective medium for surveillance applications due to an image's ability to accurately reflect the environment. Single camera streams are generally easy to intepret when viewed from a remote location, but it can be difficult to resolve multiple non-overlapping streams, especially if the observer is not closely familiar with the target environment. Furthermore, events may occur that are not visual in nature, yet may help provide additional context or may be used to generate warnings or triggers. For example, extrememly high temperatures near a gasoline storage facility may provide warning of an impending explosion, or post-explosion, data collected from temperature, humidity, and wind sensors may help discern the source or environmental context that caused the event.

1.3 Previous work

A common method of fusing multiple streams of video data is to feed individual camera streams to separate display devices as done in "security surveillance" offices. While there may be some logic to the positioning of the cameras and connection to displays, the control operator is not provided a seamless or immersive experience, and it may therefore be difficult to resolve the spatial relationship between objects in different camera views. Another approach to video fusion is to stitch the video images into a seamless 2D representation of the world. This is frequently performed in the static image domain [19][1] but systems for panoramic video acquisition from multiple cameras in pre-determined configurations have also been explored [13]. The context aware map [6] fused 2D static and video imagery of remote locations to provide visual context awareness to users. In each of these cases, an operator may navigate the 2D space which clearly enhances the connection between cameras, but as the data is inherently 2D, the operator has limited control over the viewing angles and virtual vantage points.

While significant advances have been made in image based rendering and other 2D video fusion methods the greatest flexibility is achieved with a complete 3D model of the target environment. A 3D representation has several advantages including the ability to view from arbitrary vantage points, rapid embedding of virtual models, and seamless intuitive traversal of a scene.

Achieving realism in a 3D environment requires significantly more effort than simply turning on a video camera. Hybrid approaches have therefore been explored where a rough estimate of 3D geometry is acquired and registered video is streamed onto the model's surface as a projective texture [14]. These methods merge the viewing freedom of full 3D models with the ease of detail acquisition in video streams. However, while texture projection methods provide a useful estimate of the scene's visual properties, they are insufficient as a "context awareness interface" which requires higher level semantic understanding of the environment's properties such as object locations and sizes.

To properly anchor dynamic events and objects in a 3D

environment, we must determine their location in real-time and in 3D. There are several 2D vision-based person trackers [2][18] [9][21][10], however, in a general setting with unrestricted human activities, 3D person tracking provides a much richer description for event detection [12][5]. Most existing vision trackers operate on rectilinear camera images. Omnidirectional camera arrays, however, provide the system with wide overlapped coverage of the interested space with a small number of sensors. The real-time, 3D, omni-video-array tracking method in this paper [7] is a unique setup to meet the requirements of an indoor context analysis system. For outdoor tracking, we utilize existing trackers developed in our lab.

Collaborative telepresence and visual fusion environments have been a subject of active study in recent years and include applications such as 3D teleconferencing, remote instruction, and surveillance [11]. The AVE fusion environment has integrated person and car trackers in LiDAR derived 3D environment models, and can output video of those objects in the correct 3D location [14]. A fundamental problem in collaborative environments is the access to the interaction interfaces. The steerable interfaces work by Pingali et. al. addresses the complex issue of context driven delivery of interfaces to users for ubiquitous computing: an alternative to wired or self-portable interfaces [15]. An excellent summary of context awareness work by Dey et. al. appears in [3]. They present key definitions of concepts and abstractions of computational modules related to context awareness systems as well as the Context Toolkit, a networked information sharing architecture on top of which many context awareness applications can and have been constructed. The Context Toolkit serves largely the same functionality as the NeST architecture in our current infrastructure.

Design and development of intelligent and context aware spaces is a major focus of our research at the Computer Vision and Robotics Research lab where we have considered both indoor [20][7] as well as outdoor spaces [6].

1.4 Paper outline

The work presented in this paper focuses on providing environmental context to sensor data by embedding the data into a 3D virtual world designed to mimic the environment of the physical sensor. The Networked Sensor Tapestry (NeST) is the core data acquisition software infrastructure consisting of a centralized server and modular client connectivity layer. The NeST architecture and its archiving and querying modules will be briefly described in section 2. The Context Visualization Environment (CoVE) is a graphical end-user system for visual fusion and processing of sensor data and interaction with arbitrary sensor hardware. Section 3 will focus on the CoVE and its relationship to the key components of our larger context awareness infrastructure. Currently limited context extraction is performed on the input data, but provisions for embedding research on higher level context analysis is provided architecturally by the Context Analysis Toolbox (Section 3.4) and the data sharing capabilities of the NeST. Historical analysis and context extraction can also be performed on data archived through the NeST. The paper concludes in Section 4 with a set of example applications we have built on top of the context awareness test-bed are shown in Figure 1.

2. THE NEST

2.1 Architecture

A major focus in the CVRR lab is the development of a real-time, interactive, rapidly deployable context awareness testbed. Central to the system is the data collection, archiving, and processing architecture called the Networked Sensor Tapestry (NeST) shown in Figure 1. This section summarizes the major components of the NeST, but more details can be found in [4].

The server component of the NeST allows arbitrary hardware and software clients to connect and securely share information with eachother. These clients may be *collectors* designed to harvest raw data from the environment (temperature sensors, accelerometers, cameras, etc), *semantic processors* used to extract higher level semantics from raw data feeds (object trackers, behavior analyzers, etc), or *data fusers* designed to merge streams of data from potentially correlated sensor sources. These and other clients may connect wirelessly over 802.11 or via wired ethernet. The context awareness infrastructure is malleable so as to be easily applied to a variety of applications.

2.2 Sensors

We are in development of a general purpose, extendable, modular interface device for collection and transmission of environmental sensor data using the NeST. New sensors can be seamlessly added to the device and their capabilities recognized and data shared with the NeST. The interface hardware is responsible for data acquisition from one or more attached sensors and relaying this data to the NeST via a TCP/IP connection. The hardware must also respond to requests from other applications or the server itself for changes in sensor parameters.

A variety of sensors are currently interfaced to the NeST. A TinyOS-based microcontroller (DSTINIm400) by Dallas Semiconductor [16] interfaces between a 1-wire sensor network and a (currently) wired Ethernet connection to the NeST or may connect wirelessly through an 802.11b connection on an HP Ipaq PDA. The 1-wire sensor network facilitates the addition of new sensors to the NeST. We have added a variety of 1-wire devices to the network including temperature sensors, data loggers, and timers. Analog sensors that are not 1-wire compatible may be interfaced using a 1-wire A/D converter as we have done with the Memsic accelerometer. GPS and RFID sensors can be added directly to the microcontroller via direct serial connection.

Each of these data sources provides a stream of data that is accessible to the CoVE via the NeST server architecture.

2.3 Archiving

A data archiving client constantly monitors the incoming sensor and event data and logs the information to a relational database. This information can be queried directly by clients or indirectly through context processing modules. An application (or sensor client) must describe the format of the data it provides. This description provides a template for the table structure and enables automatic population of the database, even for unrecognized clients. The archived data is used by the CoVE for replaying and analysis of past events and will be described in greater detail in Section 3.5.

Figure 2: (left) Photograph of actual SERF building. (right) CoVE virtual model.

2.4 Connection to the CoVE

The CoVE utilizes the Networked Sensor Tapestry (NeST) architecture as a feed for context data from sensors, trackers, and other applications. The CoVE connects as an application client and, like other semantic processing clients, has the ability to insert "virtual sensor" data into the context awareness environment to be used by other clients. The details of CoVE-NeST communication will be discussed in section 3.2.

3. THE COVE

The CoVE is a fundamental component of the context awareness environment providing a visual interface and fusion framework for multiple modes of sensory information. The CoVE is:

1. an interactive 3D environment for visual fusion of sensor data and embedding of information into an accurate 3D replica of the world both in real time and from archived data

2. an interactive interface to sensor control infrastructure providing instantaneous visual feedback

3. a context processing unit for extracting higher level semantic information from fused data.

This section covers the model generation method, communication with the NeST, handling of sensor data streams, replay of archived data, and the context extraction functionality integrated into the CoVE.

3.1 Model generation

Accurate context awareness requires an accurate relationship between tracked object data and the virtual environment. The accuracy of the 3D model is therefore fundamental for correct placement of objects in the scene. We have constructed a model of the SERF building and surroundings covered by our DIVA and NeST surveillance infrastructure (Figure 2). The model was created manually using a custom modeling tool and building blueprints and measurements as reference. Texture mapping was used extensively to enhance the realism of the representation while maintaining low geometric complexity. Future spaces that we wish to model can incorporate automatic model generation techniques.

3.2 Communication with the NeST

The base model is important for providing visual context to the user, but the data delivered to and handled by the CoVE is of primary interest. Establishing and maintaining

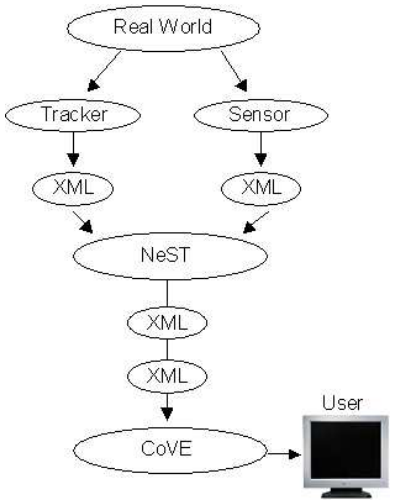

Figure 3: Flow of information from events ocurring in the real world to the CoVE.

a constant flow of information from the real world through the NeST is therefore a fundamental task.

The CoVE establishes a connection to the NeST and makes a request for the type of information it would like to receive. The NeST responds by forwarding any type of information that matches the CoVE's criteria. The NeST architecture abstracts away details of how to communicate with each individual source of information and enables the CoVE to focus on the data that is being received and in turn how to represent that data. This facilitates scalabilitiy of the CoVE from a developer's perspective, allowing the developer to easily add new input sources.

Information flows from the NeST to the CoVE in the form of XML data packets (Figure 3). An data packet consists of a description of the origin and type of information in the packet, the time the data was collected (or the event occured), and the data itself. This packet is forwarded to the CoVE through the NeST.

The XML packet shown below is a typical data packet transferred from the NeST to CoVE describing information from a person tracker:

```
<NESTMESSAGE LOCALTIME="1087933127052"
             SENDER="SENTRY_APP_INTERFACE_CLIENT"
             RECEIVER="SERVER">
  <SENSORDATA TYPE="TRACK"
              EVENTTIME="1087933127052"
              PRIVACY="PRIVATE">
    <OBJECTID ID="1"/>
    <BBOX TOP="16"
          LEFT="333"
          BOTTOM="38"
          RIGHT="350"/>
    <LOCATION3D X="-5.206172"
                Y="6.000000"
                Z="-145.666794"/>
  </SENSORDATA>
</NESTMESSAGE>}
```

When the CoVE receives one of the packets it parses the xml for the type of data. In the xml packet of Figure 2

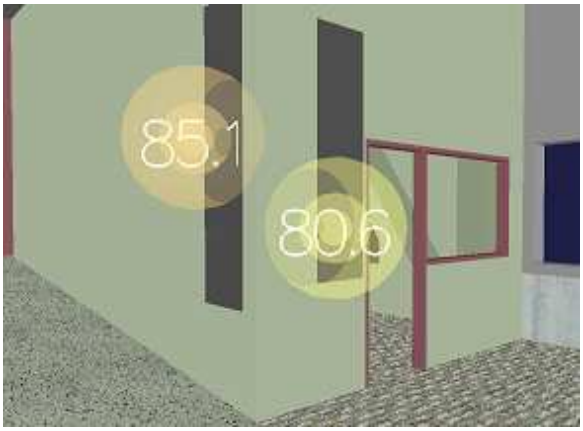

Figure 4: Visualization of temperature sensors.

track data is being received. The packet is then further processed to pull out the location of the tracked object (LOCATION3D) and the ID number the tracking program has assigned to it. If the ID number is new then an avatar is created and placed in the CoVE model corresponding to the location coordinates in the packet. If the object ID has been processed before, then the location of the avatar is updated with the information in the new packet. A similar process occurs to draw out information from sensors. Because the process of attaining data from sensors and trackers is standardized and made simple by the NeST it is easy to add new inputs into the CoVE.

3.3 CoVE Data: Sensors and Video

The goal of the CoVE is to enhance the output of sensors, video, and "virtual sensors" such as trackers by embedding it in the visual context of the actual environment in which it was acquired. This section focuses on raw data streams from sensors such as thermometers, accelerometer, and omni/rectilinear video cameras. Trackers will be discussed in more detail in section 3.3.3.

3.3.1 Sensors in the CoVE

Tracking information is not the only type of data to be visualized in the CoVE. The modular approach that has been described to visualize tracking can be extended to visualizing sensors. For instance, temperature, humidity, and seismic sensors all provide rich content that cannot be measured from a video stream. The CoVE is innately setup to offer visualizations of these other modalities. As with tracking programs that send their information to the CoVE, sensors can communicate with the NeST server and forward their data to the CoVE. When the data arrives at the CoVE, it draws a representation for that sensor in the virtual environment, and updates the visual parameters as new information is received. After a representation for the sensor has been created, any number of that type of sensor can be deployed and visualized in the CoVE.

Figure 4 shows an example of temperature sensors in the CoVE. A temperature sensor is represented by three spheres in the virtual environment at the location of the actual sensor. These spheres change color over time according to the value of the real world sensor. Also, the value is displayed in the middle of the spheres. A heat scale is used with a color gradient starting at blue indicating cold and ending at red indicating hot temperatures. Each temperature sensor is connected to the network and sends updates of its current value every two seconds.

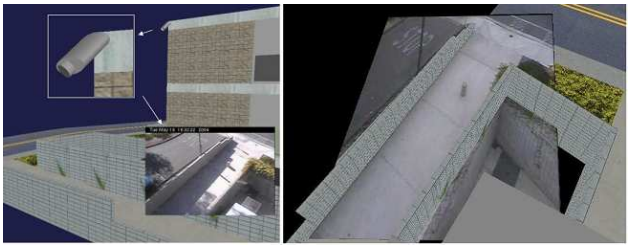

Figure 5: (left) Picture-in-picture live viewing of video streams. (right) Embedded mode with live video stream registered to 3D environment.

3.3.2 Live Video

Ideally the entire environment would be maintained as a 3D model. Unfortunately, because typical scenes are large and dynamic it is impossible to update all of the subtle nuances of the model to capture detail with the same ease as a video stream. Furthermore, there will likely be visual events such as subtle human behaviors that are not encapsulated in the simple positional information delivered by the trackers.

We currently have several video cameras distributed in the physical world inside and surrounding the SERF buiding. These video streams are used in the CoVE to enhance level of detail of monitored environments in the 3D CoVE environment. Video can be displayed in two modes: embedded and picture-in-picture.

The embedded video mode is useful for a user who needs to continuously monitor a specific neighborhood. Given the position and orientation of the camera the video footage is projected onto the surface of the 3D environment model at the correct 3D location, for example, in figure 5 live video of the walkway is projected onto the surface of the ground geometry. These streams may be enabled at the user's discretion. The picture-in-picture mode is useful for a user who wishes to watch a live view of one location in the scene, while monitoring sensors or other inputs in a separate area of the environment (see Figure 5). Also, in the picture-in-picture mode the user can control the orientation of the camera by clicking on different areas of the video stream. The zoom of the camera can also be adjusted by the user.

3.3.3 Tracking and face capture

In an indoor environment, the occupants are detected and tracked by a real-time 3D networked omni-video array (NOVA) tracker [7] operating on four omni-cameras. The 3D tracker provides location, velocity, and height information of the occupants. The horizontal locations of people are then triangulated by n-ocular stereo [17] and in turn are used to compute the heights of people from the topmost pixels with camera calibration parameters. Each measurement is then either associated to the existing tracks or initiates new tracks. Continuously unassociated tracks are terminated. Each track is estimated by a constant velocity Kalman filter.

Given the tracks extracted using the NOVA tracker, active camera selection determines an appropriate PTZ camera to

capture the face of person. If the person is static, the closest omnicam is selected; otherwise the PTZ camera that faces the walking person is selected. The directionality is defined by the inner product of the walking velocity with the vector from the person to a camera. The best camera is chosen to have the largest inner product. Then pan, tilt, and zoom values are determined by the relative direction and distance of the head of the person to the best camera. The person's face is detected and cropped by a face detector utilizing skin-tone segmentation and elliptical edge detection as the detection features [8].

3D trackers are a critical source of context information for the CoVE. The NeST architecture allows the CoVE to view trackers as "black boxes". The CoVE extracts object ID, location, and size from the tracker data packets and instantiates a 3D avatar at the appropriate location in the virtual environment. The 3D visualization of track data in the modeled space provides significantly better intuition than viewing the raw 2D tracks alone. Using the height estimates to control the size of the avatar an operator can immediately compare sizes of different tracked objects.

3.4 Context Analysis Toolbox

Because the CoVE maintains an accurate 3D record of the environment including the building geometry and key static and dynamic objects within the modeled space, it is in a unique position to extract additional contextual information from the scene via the Context Analysis Toolbox.

The toolbox consists of a variety of higher order processing plugins that can analyze real-time or historical data to extract semantics and statistics from the incoming environment data. Current toolbox modules include a *zone alarm*, where regions in the 3D world may be specified and the number of people entering this zone calculated. The current functionality can be easily extended to calculate behavioral statistics such as a *crowd size calculator* that determines the number of individuals in a given region.

The CoVE (and hence the Context Analysis Toolbox) are connected to the context archive. This allows a user perform queries such as searching for activity by time, person ID, or if the appropriate object and location labels are provided, contextual pointers such as recalling all subjects who entered through a main door, or who sat around a table.

3.5 CoVE Live Mode vs. Replay

There are two modes of operation for the CoVE: Live and Replay mode. The traditional Live mode, used for real-time monitoring of surveillance sensors, utilizes data streamed directly from the NeST in real time. However, dealing with historical data is critical for post-mortem analysis of events in the sensed environment. In replay mode, the CoVE connects directly to the surveillance archive created and maintained by the NeST and submits queries to define the specific data to "replay" in the environment.

Any query can be submitted to the database to extract context data, but currently an interface is provided to retrieve sensor data within a given range of time and date. Replay data can be fed into the system as though it was live for re-visualization purpose but the frame rate of the visualization system is often faster than the sample rate of the sensors. The records retreived from the database are keyframe animated using the timestamp field of the database, with an option for bilinear interpolation to smooth the animation.

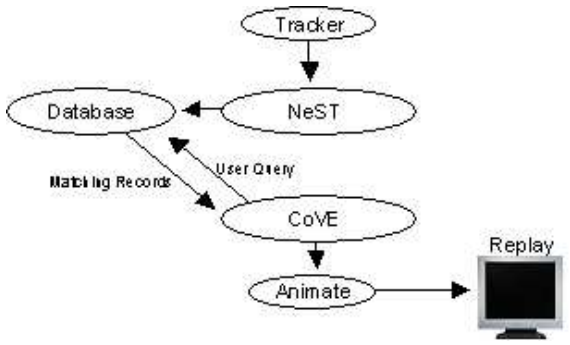

Figure 6: Flow of information to the CoVE in replay mode.

Replay data can also be processed further by the Context Analysis Toolbox, for example, to extract the number of people occupying a room within a given time window as described in section 3.4.

Figure 6 shows the flow of data during the replay mode. First, the user selects a time period to observe from the Replay Control Panel. Then the database is queried for all events that occurred during that time period. The returned data is then separated into groups of each unique object (for example, each unique object that was found by a tracker).

Objects are animated using keyframe animation. Each returned record is saved as a keyframe which will be used to determine where the object is located during the playback. At each timestep during playback, the CoVE traverses the list of objects and tests for their presence in the scene. If the object is present at the current timestep, then its position is determined by the interpolation between the keyframes before and after the current replay time.

4. ILLUSTRATIVE EXAMPLES

The flexibility of our context awareness infrastructure has allowed rapid construction and deployment of several experimental applications. Two 2D trackers have been integrated for outdoor video surveillance in addition to the NOVA tracker for indoor 3D tracking. This section showcases a selection of applications we have deployed.

4.1 Indoor: Person tracking and face capture

3D tracking with avatar generation. Figure 7 shows 2 people tracked in an indoor lab space using the NOVA tracker. The tracker analyzes four omni directional cameras to extract 3D location and height estimates of each of the objects. A box is placed in the virtual model at the estimated location with its height matching the height of the tracked object.

Selective Focusing. The selective focusing ability of the NOVA tracker allows us to embed live video of the tracked person's face onto their avatar in real-time. NOVA shifts the cameras to the 3D location of the subject and if multiple subject are present, identifies which subject in the scene is the focal point. The face is detected and the "best view camera" is zoomed onto the face region. The best view is defined by the closest camera that detects a frontal face in the image. The CoVE then acquires this video in real time, and streams the data onto the surface of the avatars (Figure 8).

Face Cropping and ID Recall. Each tracked subject is assigned a unique global identifier by NOVA. When a subject appears at a "good" vantage point, the camera with the best view automatically pan, tilts, and zooms to capture a close up video. This becomes the input to the head pose estimation and face detector modules, which in real-time, outputs a cropped face video stream. The face verification and face recognition modules further process the cropped face video, once again in real time.

NOVA can insert the image into the context archive with the time stamp, indexed by the same global object identifier as the object's track. This image can then be recalled from the database by the CoVE and attached to the track data to provide a static identifier. This is especially useful during replay of historical information from the database. The ID recall capability is illustrated in Figure 9. The units of the database time-stamp are "milliseconds since January 1, 1970".

For the recall to be performed automatically, the track ID stored in the archive must be automatically associated with the person tracked in real-time. This requires some form of object verification (ie, face recognition) currently not present in the tracker. In the current implementation, the subject images are therefore manually inserted into the database and the tracks are manually associated with the database images. The addition of object verification is near-term future work.

Zone Alarm. We can also define 3D watch zones in the CoVE that, when crossed by a tracked person, generate an alarm. These intrusions can be queried from the database and statistics computed

The Zone Alarm mode is shown in figure 10. The blue boxes are watch zones defined by a user in an xml packet. The boxes turn red when a tracked subject enters the zone. This alarm can be handled locally, fed back to the NeST for other applications to respond to the event, or inserted directly into the context archive.

4.2 Outdoors: Perimeter Monitoring and Notification

Outdoor tracking with CoVE context embedding. Outdoor tracking clients are connected to the NeST architecture and their output embedded in the CoVE visualization environment in the same manner as the raw NOVA tracking data. Figure 11 shows the results of one tracker tracking two subjects along a walkway. The tracker results are shown to the right of the visualization.

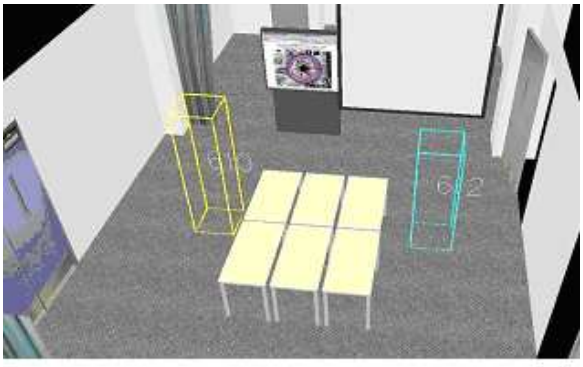

Figure 7: Objects are tracked in the real world and dynamic avatars take their place in the virtual environment.

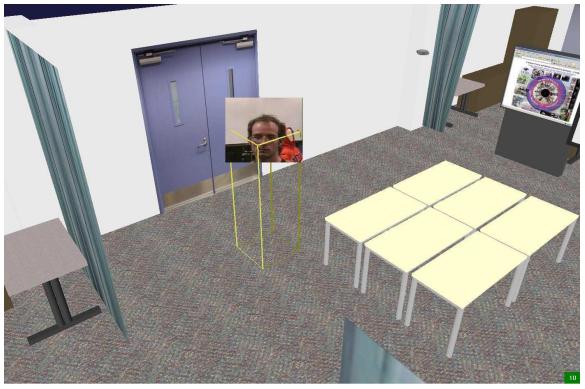

Figure 8: Video captured from a camera in the NOVA array is streamed onto the tracked subject's avatar. The "best view" camera is automatically selected to capture the frontal face.

IDENTITY TABLE

IMAGE	ID	NAME	TIME STAMP
	0	Doug	1082675273150
	1	Pew	1082675293557
	2	Kohsia	1082675301682
	3	Mohan	1082675314275
	4	Stefan	1082675330432

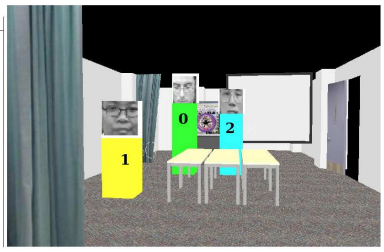

Figure 9: Archived images for individuals recognized by the NOVA tracker are recalled and embedded.

Sentry. A perimeter sentry application allows a user to specify a location of interest or watch zone in a video stream. Objects are tracked through the scene and a trigger is generated if the perimeter is breached. This breach event is sent to the NeST and subsequently forwarded to the CoVE and other context processors. Figure 12 shows the perimeter sentry client application with a zone defined in the image plane.

Behavior triggers. A context procesing module has been constructed to flag running individuals based on their travel velocity. Figure 13 shows an example of the running behavior triggering a switch from a solid box shown in Figure 11 (blocking the embedded video stream) to a wireframe box. More complex behavior triggers can be defined and easily plugged into the NeST.

5. CONCLUSION AND FUTURE WORK

The NeST, CoVE, trackers and sensor hardware form a powerful and flexible context awareness testbed. The prototype infrastructure in place allows rapid reconfiguration and deployment of a variety of classes of applications including surveillance, remote collaboration, and context awareness. We hope to leverage this infrastructure to assemble experimental applications and context processing research modules in a deployable real-world environment.

Figure 10: Watch zones are defined at entry ways shown in blue. Red indicates a tracked subject has passed through a watch zone.

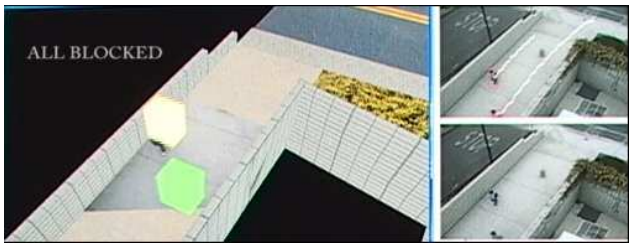

Figure 11: Outdoor tracking data embedded in CoVE.

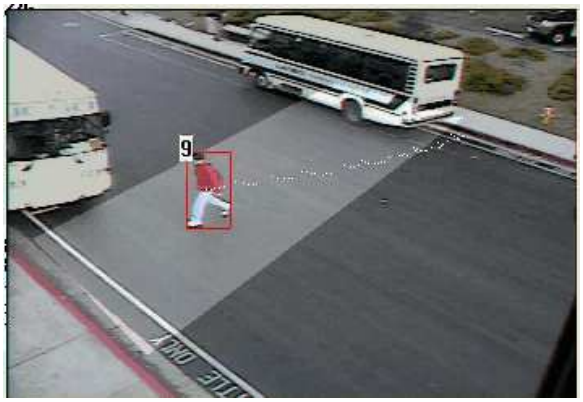

Figure 12: The watch zone is defined by the lightened area in the image on the street. The box around the tracked figure changes from green to red as the zone is breached. This event is sent to the NeST, archived and/or forwarded to the CoVE.

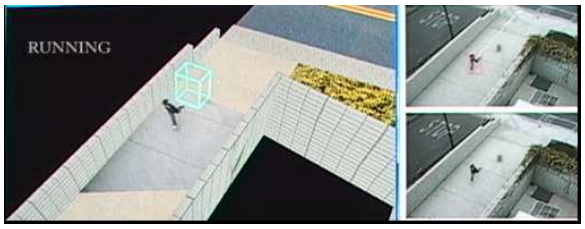

Figure 13: Outdoor tracker with a running behavior trigger. The CoVE responds to the trigger by changing the box from solid to wireframe.

The NeST architecture is a combination of custom software middleware and data collection hardware that provides an abstraction of data and processors and serves as the physical and conceptual glue for our context awareness systems. The NeST provides data collection, sharing, archiving, and filtering capabilities and serves this functionality to the CoVE and other client applications. Core abstractions have been developed in the NeST server architecture for applications to share data and for this data to be archived and queried for further processing by context extraction units. Context extraction can be directed by a user or supervising application using any number of plugins from the Context Analysis Toolbox. Though the toolbox is currently sparsely populated, it provides a clean conceptual interface for experimentation with context extraction from both real-time and historical sensor data.

The CoVE provides a visual interface and fusion framework for multiple modes of sensory information. Visual and non-visual sensor data is merged and accurately registered in the CoVE and visualized in a seamless 3D replica of the sensed environment. Monitoring and analysis of the changing context can occur in real time via data streamed through the NeST or can be performed on historical data from the context archive. A user can also actively interact with sensors to change properties such as the PTZ parameters of an embeddad camera and receive instantaneous visual feedback.

This is version 1.0 of the context awareness environment. We intend to use the experiments developed atop this infrastructure to define new abstractions and key components for subsequent revisions of the architecture. Currently the definition of the 3D world (object labels, geometry, and locations) are maintained internally by the CoVE application. By populating the context archive with this information, we can perform queries linking the environment to the events occuring in it. The current system is limited to contextualization of data sources, and doesn't handle the processing, rendering, or extraction of higher-level context. The context analysis toolbox is therefore a the focal point for future development. We are also interested in mobilizing the data collection infrastructure and visualization environment.

6. ACKNOWLEDGEMENTS

Our research is supported in part by the Technical Support Working Group (TSWG) of the US Department of Defense and the UC Discovery Program. We are thankful for contributions of our colleagues from the Computer Vision and Robotics Research Laboratory in the design and development of the experimental testbeds and that of Dr. Tarak Gandhi for his valuable assistance with the outdoor tracker modules and Kohsia Huang for helping to connect the NOVA tracker to the NeST and CoVE. We also thank our reviewers for their many thoughtful comments and suggestions.

7. REFERENCES

[1] S. E. Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Computer Graphics (SIGGRAPH 95)*, pages 29–38, August 1995.

[2] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenswalb. Plan-view trajectory estimation with dense stereo background models. In *Proceedings of the International Conference on Computer Vision*, 2001.

[3] A. K. Dey, D. Salber, and G. D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction Journal*, 16(2-4):97–166, 2001.

[4] D. A. Fidaleo, H. Nguyen, and M. M. Trivedi. The networked sensor tapestry (nest): A privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In *ACM 2nd International Workshop on Video Surveillance and Sensor Networks*, New York, New York, October 2004.

[5] D. Gavrila and L. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, CA, June 1996.

[6] B. Hall and M. Trivedi. A novel interactivity environment for integrated intelligent transprotation and telematic systems. In *5th International IEEE Conference on Intelligent Transportation Systems*, pages 396–401, Singapore, September 2002.

[7] K. S. Huang and M. M. Trivedi. Video arrays for real-time tracking of person, head, and face in an intelligent room. *Machine Vision and Applications*, 14(2):103–111, June 2003.

[8] K. S. Huang and M. M. Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video streams. In *To appear in the Proceedings of International Conference on Pattern Recognition*, Cambridge, UK, August 2004.

[9] I.Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[10] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 2000.

[11] J. Leigh, A. E. Johnson, M. Brown, D. J. Sandin, and T. A. DeFanti. Visualization in teleimmersive environments. *Computer*, 32(12):66–73, 1999.

[12] I. Mikic, S. Santini, and R. Jain. Tracking objects in 3d using multiple camera views. In *ACCV 2000*, Taipei, Taiwan, January 2000.

[13] U. Neumann, T. Pintaric, and A. Rizzo. Immersive panoramic video. In *Proceedings of the 8th ACM International Conference on Multimedia*, pages 493–494, October 2000.

[14] U. Neumann, S. You, J. Hu, B. Jiang, , and I. O. Sebe. Visualizing reality in an augmented virtual environment. *Presence:Teleoperators and Virtual Environments Journal*, 13(2):222–233, April 2004.

[15] G. Pingali, C. Pinhanez, A. Levas, R. Kjeldsen, M. Podlaseck, H. Chen, and N. Sukaviriya. Steerable interfaces for pervasive computing spaces. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 315. IEEE Computer Society, 2003.

[16] D. Semiconductor. http://www.maxim-ic.com/TINIplatform.cfm.

[17] T. Sogo, H. Ishiguro, and M. M. Trivedi. *Panoramic Vision*, chapter N-Ocular Stereo for Real-Time Human Tracking, pages 359–375. Springer-Verlag, 2001.

[18] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):745–757, August 2000.

[19] R. Szeliski and H.-Y. Shum. Creating full view panoramic mosaics and environment maps. In *Computer Graphics (SIGGRAPH 97)*, pages 251–258, August 1997.

[20] M. M. Trivedi, K. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *IEEE Transactions on Systems, Man, and Cybernetics, special issue on Ambient Intelligence*, July 2004.

[21] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.