RealityFlythrough: Enhancing Situational Awareness for Medical Response to Disasters Using Ubiquitous Video

Neil J. McCurdy, MS[1], William G Griswold, PhD, and Leslie A. Lenert, MD[1,2]
[1]California Institute for Telecommunications and Information Technology,
University of California, San Diego, La Jolla, CA
[2]Veterans Affairs San Diego Healthcare System, San Diego, CA

The first moments at a disater scene are chaotic. The command center initially operates with little knowledge of hazards, geography and casualties, building up knowledge of the event slowly as information trickles in by voice radio channels. RealityFlythrough is a tele-presence system that stitches together live video feeds in real-time, using the principle of visual closure, to give command center personnel the illusion of being able to explore the scene interactively by moving smoothly between the video feeds. Using RealityFlythrough, medical, fire, law enforcement, hazardous materials, and engineering experts may be able to achieve situational awareness earlier, and better manage scarce resources. The RealityFlythrough system is composed of camera units with off-the-shelf GPS and orientation systems and a server/viewing station that offers access to images collected by the camera units in real time by position/orientation. In initial field testing using an experimental mesh 802.11 wireless network, two camera unit operators were able to create an interactive image of a simulated disaster scene in about five minutes.

Figure 1. Hazardous materials MMRS team member suiting up with a gasmask mounted video camera for broadcasting of images during

similar problems with situational awareness (2) and as a result, leaders believe that live video broadcasted into the command center could be of benefit.

RealityFlythrough (3) is a tele-presence system that stitches together live video feeds in real-time, giving command center personnel not only the ability to view video of the scene but also the ability to explore the scene interactively by moving smoothly between the video feeds. Using RealityFlythrough, medical, fire, law enforcement, hazardous materials, and engineering experts can obtain situational awareness much earlier than is currently possible. The contribution of this paper is a feasibility study. We investigate if it is possible to use RealityFlythrough in the harsh conditions of a disaster where no assumptions about the location of the disaster or existing infrastructure (e.g. computer network) can be made, and we identify the requirements for successful deployment. We then report the results of using a prototype of RealityFlythrough in a recent drill designed to evaluate its performance in a field setting comparable to that of a disaster site.

INTRODUCTION AND BACKGROUND

The first moments at a disater scene are chaotic. The command center for first responders, once established, initially operates with limited information, building up knowledge of the event hazards and the severity of injury of victims, as slowly as information trickles in. When disasters are caused by terrorist attacks with chemical, biological, and radiological weapons, situtational awareness is critical to safe operations. Operations cannot proceed unless there is sufficient situation awareness to limit the hazards to acceptable levels. Loss of awareness or the failure to achieve adequate awareness can have catastrophic outcomes (1).

In disaster drills performed by the San Diego Metropolitan Response System teams (Regional teams supported by the Department of Homeland Security and trained to respond to events involving weapons of mass destruction) commanders have had

DESIGN CONSIDERATIONS

RealityFlythough addresses the problem of

Figure 2. Snapshots of a transition. The transition uses two "filler" images to provide additional contextual information. During this transition the viewpoint moves roughly 20 meters to the right of the starting image and rotates 135 degrees to the right.

integrating multiple moving video feeds over time. There are a number of different ways to present multiple video sources to a user. Security guards have been using large arrays of monitors for years. Although it may take awhile for the guards to learn the camera-to monitor mappings, the system works as well as it does only because the mappings rarely change. The same setup would be less useful in a disaster environment for two reasons:

(1) The camera-to-monitor mappings would have to be relearned at each new environment, delaying the users' ability to acquire situational awareness, and
(2) The system deals poorly with head-mounted cameras that move with the wearer, making it nearly impossible to associate a monitor with a particular location. With little correlation between the position of the video displays and the real location of the cameras, confusion and bad decisions, especially in high stress environments, will be the likely outcome.

An alternative approach is to capture the location and orientation of each camera and display the camera position on a map of the scene. The video feeds can then be situated in space, allowing a user to select the optimal view by simply choosing a position on a map. Even this system is not adequate, however. A great deal of cognitive effort is required to translate between the immersive 3d environment of a video display and the 2d bird's-eye representation of the scene. This task is difficult enough when dealing with static cameras that have a fixed location, and is much more complex when both the source and destination cameras are moving and rotating.

An approach that minimizes the cognitive effort required to understand the relative camera positions is to dynamically stitch together the images into a complete view of the environment. This gives the users a sense that they are in the environment—an experience similar to that provided by first-person immersive video games (e.g. Doom) where users can interactively move through the scene choosing the camera angles that best suit their needs.

Since the area of a disaster scene may be large relative to the number of cameras available, software systems would need to capture and save individual still images as the cameras move through the environment. This not only gives the command center access to the most recent image at a particular location, but it also provides additional contextual information to the users as they move through the virtual representation of the scene. If the live video cameras are located far apart or are facing opposite directions, there will be large gaps between the images. In situations where the timeliness of the data is not critical, the users could choose to fill these gaps with the archived imagery to give them a better sense of the spatial relationships between the cameras. A clear indication of the age of the data would need to be displayed.

It may also be useful to be able replay earlier events to review information that recently occurred, to review material during post-mortem analysis, or to aid with future training exercises. To support this, a system should archive the video streams and allow PVR-style (personal video recorder) time-shifting across all streams.

IMPLEMENTATION

RealityFlythrough is in part an exercise in the design of an experimental user interface for three-dimensional data. The design focuses on the creation of an imperfect, but sensible and natural enough illusion of three-dimensional movement through space that provides the necessary contextual information for user orientation in space without requiring much conscious thought. The interface design assumes that an approximation of relationships between images in three-dimensional space is enough to allow the human visual system to infer relationships between the images. The approach is based on the assumption that the brain is adept at committing closure—filling in the blanks when given incomplete information (4). Visual closure is a constant in our lives; closure, for example, conceals from us the blind spots that are

present in all of our eyes. Because Reality Flythrough is designed to take advantage of visual closure, the effect of the system is best demonstrated by watching a video presentation illustrating the illusion (5). Figure 2 attempts to illustrate the process with still pictures but does not achieve the illusion.

RealityFlythrough works by situating 2d images in a virtual 3d environment. Since the position and orientation of every camera is known, a representation of the camera can be placed at the corresponding position and orientation in virtual space, but because of visual closure, absolute accuracy is not a requirement. The camera's image is then projected onto a virtual wall (see Figure 3). When the user is looking at the image of a particular camera, the user's position and direction of view in virtual space is identical to the position and direction of the camera. As a result, the entire screen is filled with the image. Referring to Figure 2, a *transition* between camera A (the left-most image in the figure) and camera B (the center image in the figure) is achieved by smoothly moving the user's position and view from camera A to camera B while still projecting their images in perspective onto the corresponding virtual walls. The rendered view situates the images with respect to each other and the viewer's position in the environment. OpenGL's standard perspective projection matrix is used to render the images during the transition, and an alpha-blend is used to transition between the overlapping portions of the source and destination images. By the end of the transition, the user's position and direction of view are the same as camera B's, and camera B's image fills the screen.

An example may make it easier to understand how RealityFlythrough works. Imagine standing in an empty room that has a different photograph projected onto each of its walls. Each image covers an entire wall. The four photographs are of a 360 degree landscape with one photo taken every 90 degrees. Position yourself in the center of the room looking squarely at one of the walls. As you slowly rotate to the left your gaze will shift from one wall to the other. The first image will appear to slide off to your right, and the second image will move in from the
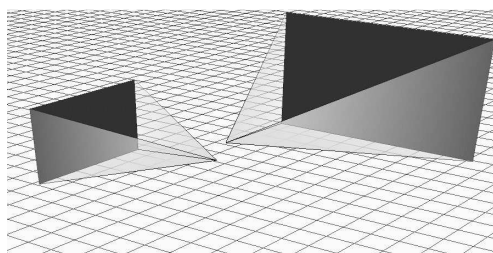
left. Distortions and object misalignment will occur at the seam between the photos, but it will be clear that a rotation to the left occurred, and the images will be similar enough that sense can be made of the transition. Reality Flythrough operates in a much more forgiving environment: the virtual walls are not necessarily at right angles, and they do not all have to be the same distance away from the viewer.

Image manipulation
The method used for displaying the images obviates the need for image stitching. Stitching is a computationally expensive process that requires very precise knowledge of the locations of the cameras and the properties of the lenses. The discussion section considers related work that uses techniques like stitching, but in more constrained, artificial environments.

The challenge with doing on-the-fly "stitching" is in selecting which images to display at any given moment. If all available images were displayed simultaneously, the resulting jumble would be unintelligible. What RealityFlythrough does is select the most optimal video feed to display at every position by considering such fitness parameters as the camera's proximity to the user's virtual position, the percentage of the user's view that the feed would consume, and the recency and liveness of the camera source. Since RealityFlythrough can display archived photos to fill in gaps during transitions, the recency metric selects for the most recent of these images, and the liveness metric prefers live video feeds over archived photos.

Displaying the optimal photo at every point in space is not sufficient. In order to realize the benefit of visual closure, the users have to have time to process the imagery. Transitions that have a duration of at least one second do well in practice, so RealityFlythrough looks ahead one second during transitions, estimating the user's position and selecting the most optimal camera for that position. The result is a smooth transition that is both sensible and pleasing.

Cameras and position sensing units
RealityFlythrough image acquisition units combine different off-the-shelf technologies to record images and position in space and transmit these data to a central server. Camera units have four components: a USB web camera, a WAAS GPS receiver (eTrex, Garmin, Kansas City), an EZ-Compass positional sensor that reports the tilt, roll and yaw

Figure 3. An illustration of how the virtual cameras project their images onto a wall.

movements at 15 Hz (AOSI, Linden NJ), and a Windows XP tablet computer with 802.11 wireless connectivity (HP TC1100, Hewlett Packard, Palo Alto, Ca), that serves as an integrative platform and a data communications device. Video and other sensor data is transmitted using the OpenH323 (6) implementation of the H323 video conferencing standard.

RealityFlyThrough Server and Visualization Unit
Images and position information from the acquisition units are relayed wirelessly to a central server. The server is standard IBM T42 laptop. The server receives and manages video streams using a modified MCU (Multipoint Control Unit) that was built on top of OpenH323 (6). The video is decoded, integrated with the other sensor data (position data and soon audio data), and forwarded to the RealityFlythrough engine. If the user is viewing this particular video feed, the video is rendered on the screen. Each video frame is also archived if the image quality is deemed to be good enough, with the newly archived image replacing whatever other image was taken from that position in space. Over time, the most recent archived photo will be available at every position of the disaster scene. The positions of the archived images are stored in a spatial index to speed querying during transitions.

Network
RealityFlythrough assumes the presence of a stable 802.11 data network for videoconferencing. The system is designed to work under the bandwidth limitations of the mesh-style wireless distribution network being developed for the WIISARD project.

Evaluation
To assess the ability of RealityFlythrough to function under disaster conditions, we conducted a proof-of-concept experiment in an environment that, from a technology point of view, mimicked many of the conditions of a disaster. We ran RealityFlythrough on a mesh network created with three battery-operated 802.11b Access Points (AP) using a wireless distribution system for AP to AP communications (7). The test scene was a large (roughly 90x30 meter) outdoor open space on the UCSD campus.

The live video feeds, position data, and orientation data from the camera units were transmitted over the 802.11 network to the central server while camera operators wandered around the test area. The camera operators crossed paths occasionally, but generally stayed in their respective areas. While information was being collected, an operator at the command center (NM) moved virtually around the scene to make sure the video feeds were being captured and the system was running smoothly. The video feeds, along with the position and orientation data, were archived so that the entire experience could be replayed multiple times for further analysis.

The functioning of the system was somewhat limited because of network issues. The experimental network provided adequate performance. With two cameras running, we achieved an average frame rate of roughly 4 fps, with the actual rate varying between 2 and our target of 5 fps. We saw a significant amount of packet loss (8.7% for one camera and 3.7% for the other), but the majority of these packets were lost when the cameras went out of range of their initial access point and had difficulty reassociating with another access point. Despite these difficulties, within five minutes the walkable region of the visual space was well covered by still photos allowing anyone using RealityFlythrough to quickly explore the scene.

DISCUSSION

We have introduced RealityFlythrough, a telepresence system that allows command center personnel to navigate through a disaster scene by transitioning between live video, still images, and archived content. Moving freely through video streams of the site in both time and space, the command center may be able to achieve the necessary situational awareness faster than using radio communications. We have shown that RealityFlythrough can work in the field conditions expected at a disaster scene. Future work will explore its functioning during actual deployments of the MMRS and the impact of the system on incident commander decision making.

There have been several approaches to telepresence with each operating under a different set of assumptions. Telepresence (8), tele-existence (9), tele-reality (10,11), virtual reality and tele-immersion (12) are all terms that describe similar concepts but have nuanced differences in meaning. Telepresence and tele-existence both generally describe a remote existence facilitated by some form of robotic device or vehicle. There is typically only one such device per user. Tele-reality constructs a model by analyzing the images acquired from multiple cameras, and attempts to synthesize photo-realistic novel views from locations that are not covered by those cameras. Virtual Reality is a term used to describe interaction with virtual objects.

First-person-shooter games represent the most common form of virtual reality. Teleimmersion describes the ideal virtual reality experience; in its current form users are immersed in a CAVE (13) with head and hand tracking devices.

RealityFlythrough contains elements of both telereality and telepresence. It is like telepresence in that the primary view is through a real video camera, and it is like tele-reality in that it combines multiple video feeds to construct a more complete view of the environment. RealityFlythrough is unlike tele-presence in that the cameras are likely attached to people instead of robots, there are many more cameras, and the location and orientation of the cameras is not as easily controlled. It is unlike tele-reality in that the primary focus is not to create photo-realistic novel views, but to help users to internalize the relationships between available views.

All of this work (including RealityFlythrough) is differentiated by the assumptions that are made and the problems being solved. Telepresence assumes an environment where robots can maneuver, and has a specific benefit in environments that would typically be unreachable by humans (Mars, for example). Tele-reality assumes high density camera coverage, a lot of time to process the images, and extremely precise calibration of the equipment. The result is photorealism An alternative tele-reality approach assumes a-priori acquisition of a model of the space (14) with the benefit of generating near photo-realistic live texturing of static structures. And finally, RealityFlythrough assumes mobile ubiquitous cameras of varying quality in an everyday environment. The resulting system supports such domains as disaster command and control support.

Reality Flythrough can operate in a disaster response setting because the only required input to the system other than the video feeds is the location, orientation, and field of view of each camera. No processing of the image data is necessary, so the system can stitch together live video feeds in real-time. Not processing the image data has other benefits, as well. There are no requirements on the quality or even the type of imagery that is captured. The system works equally well across the spectrum of image quality, from high-resolution high frame-rate video to low-quality still images. Enhancements of the system by integration of video with other types of imaging sources such as thermal imagery that penetrates smoke or infrared imagery that penetrates darkness are also possible. While further evaluation and additional refinements are necessary, RealityFlythrough is a promising approach to enhance the situational awareness of command staff using ubiquitous video.

ACKNOWLEDGEMENTS

REFERENCES

1. McKinney_&_Company. Post 9-11 Report of the Fire Department of New York; August 2002.
2. Domestic Preparedness Program: Operation Grand Slam - After-Action Report. San Diego, CA: San Diego County; June 2001 August 2000.
3. N. J. McCurdy and W. G. Griswold. A systems architecture for ubiquitous video. Proceedings of the Third International Conference on Mobile Systems, Applications, and Services (MobiSys 2005). Usenix (2005), 1–14.
4. S. McCloud. *Understanding comics: The invisble art.* Harper Collins Publishers, New York, 1993.
5. N. J. McCurdy and W. G. Griswold. Telereality in the wild. UBICOMP'04 Adjunct Proceedings, 2004. http://activecampus2.ucsd.edu/ ˜nemccurd/tele_reality_wild_video.wmv.
6. http://www.openh323.org
7. M Arisoylu et al.. 802.11 Wireless infrastructure to enhance medical response to disasters. Proc. AMIA Fall Symp. 2005 (under review).
8. H. Kuzuoka et al. Can the gesturecam be a surrogate? In *ECSCW*, pages 179–, 1995.
9. S. Tachi. Real-time remote robotics - toward networked elexistence. In *IEEE Computer Graphics and Applications*, pages 6–9, 1998.
10. R. Szeliski. Image mosaicing for tele-reality applications. In *WACV94*, pages 44–53, 1994
11. T. Kanade, P. Rander, S.Vedula, and H. Saito. Virtualizedreality: digitizing a 3d time varying event as is and in real time. In *Mixed Reality, Merging Real and VirtualWorlds.* SpringerVerlag, 1999.
12. J. Leigh, A. E. et al. A review of tele-immersive applications in the CAVE research network. In *VR*, page 180.
13. C. Cruz-Neira, D. Sandin, T. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In SIGGRAPH '93, pages 135-142, 1993.
14. U. Neumann et al. Augmented virtual environments: Dynamic fusion of imagery and 3d models. In *VR '03: Proceedings of the IEEE Virtual Reality 2003*, page 61, 2003.