Harnessing Mobile Ubiquitous Video

Neil J. McCurdy William G. Griswold

Abstract

We are rapidly moving toward a world where personal networked video cameras are ubiquitous. Already, camera-equipped cell phones are becoming commonplace. Imagine being able to tap into all of these live video feeds to remotely explore the world in real-time. We introduce RealityFlythrough, a telepresence system that makes this vision possible. By situating live 2d video feeds in a 3d model of the world, RealityFlythrough allows any space to be explored remotely. No special cameras, tripods, rigs, scaffolding, or lighting is required to create the model, and no lengthy preprocessing of images is necessary. Rather than try to achieve photorealism at every point in space, we instead focus on providing the user with a sense of how the video streams relate to one another spatially. By providing cues in the form of dynamic transitions, we can approximate photorealistic telepresence while harnessing cameras "in the wild." This paper describes the RealityFlythrough system, and reports on a live flythrough experience. We find that telepresence can work in the wild using only commodity hardware and off-the-shelf software, and that imperfect transitions are sensible and provide a compelling user experience.

1 Introduction

We are rapidly moving toward a world where personal networked video cameras are ubiquitous. Already, camera-equipped cell phones are becoming commonplace. Imagine being able to tap into live video feeds to remotely explore the world in real time. We introduce RealityFlythrough, a telepresence system that makes this vision possible.

There are numerous applications for such a system, but perhaps the most compelling involves disaster response. Consider, for example, first responders equipped with head-mounted wireless video cameras encountering the chaos of a disaster site. As they fan out through the site, they continuously broadcast their location, orientation, and what they see to a RealityFlythrough server. The responders' central command remotely explores the site by viewing these video feeds to get a sense of the big picture. They can then direct medics to the injured, firefighters to potential flare-ups, and engineers to structural weaknesses. As more people enter the site and fixed cameras are positioned, the naturalness of the flythrough is enhanced until ultimately the entire space is covered and central command can "fly" around the site looking for hot spots without constraints.

Other applications for RealityFlythrough range from enabling the disabled to remotely explore the world, to providing a mechanism for sports fans to remotely fly around a stadium selecting the optimal vantage point for viewing a game.

As will be discussed in detail in Section 3, there have been many approaches to creating immersive environments that promote exploration of either a remote or a virtual space. The virtual reality community builds the environments from scratch, using photograph-based texture maps if necessary and where possible; the graphics and vision communities create photorealistic renderings of novel views using photographs (and in some cases video feeds) taken from different angles; and the robotics community achieves the effect by attaching a camera to a remote-controlled robot.

Our work starts with a different set of assumptions, and as a result leads to a very different design. The goal of RealityFlythrough is to harness networked ubiquitous cameras. Ubiquitous cameras are everywhere, or at a minimum can go anywhere. They are inside, outside, carried by people, attached to cars, on city streets, and in parks. Ubiquity moves cameras from the quiet simplicity of the laboratory to the harsh reality of the wild. The wild is dynamic – with people and objects constantly on the move, and with uncontrolled lighting conditions; it is uncalibrated – with the locations of objects and cameras imprecisely measured; and it is variable – with video stream quality, and location accuracy varying by equipment being used, and the quantity of video streams varying by location and wireless coverage. Static surveillance-style cameras may be available, but it is more likely that cameras will be carried by people. Mobile cameras that tilt and sway with their operators present their own unique challenges. Not only may the position of the camera be inaccurately measured, but sampling latency can lead to additional errors.

It is a non-trivial challenge to support live and real-time remote exploration of the world. The ideal is to have a camera lense at every possible vantage point so that a photorealistic view can be realized from anywhere. Given the pragmatic limits to ubiquity, this will not be an option in the near term. The solution, then, is to take advantage of the camera lenses that are available, and to either attempt to synthesize a novel view from the available images, or to provide a mechanism for the user's view to transition from one image to another. The synthesis of photorealistic novel views in real-time is not possible with today's technology given the conditions of the wild, but it is possible to generate sensible transitions between camera feeds.

RealityFlythrough uses these transitions to convey spatial context. Transitions are a dynamic, real-time blend from the point of view of one camera to the point of view of another, and are designed to help the user generate an internal conceptual model of the space. See Fig. 1. Although it is possible to stop mid-transition to see a novel view, the emphasis is on displaying the real images captured from cameras. The transitions from camera to camera are provided mainly to help the user make sense of how the starting and ending images are related to one another spatially.

The contribution of this paper is to show that telepresence can be made to work while relaxing the constraints of a tightly controlled environment; that is, to work in the wild. We have created a compelling telepresence experience using commodity components and off-the-shelf software. With consumer GPS's, inexpensive web cameras, and standard video conferencing software, we were able to construct an environment that provides a live, immersive telepresence experience. We found that a pair of research subjects remotely exploring a physical space had a compelling experience and were able to determine key facts about the activities going on there.

In the next section we will discuss how the transitions in RealityFlythrough are achieved. After presenting our work, we will look more closely at related work in Section 3. We will then describe our qualitative flythrough study with live video and moving cameras. We discuss the qualities that made the experience a success and identify elements that require further refinement. We close with a conclusion and discussion of future work.

2 Approach

RealityFlythrough works by situating 2d images in 3d space. Because we know the position and orientation of every camera, we can place a virtual camera at the corresponding position and orientation in virtual space. The camera's image is then projected onto a virtual wall (see Fig. 2a). 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 Fig. 1, a *transition* between camera A (image (a) in the figure) and camera B (image (d) 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. By using OpenGL's standard perspective projection matrix to render the images during the transition, the rendered view situates the the images with respect to each other and the viewer's position in the environment. 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. Additional processing and imagery, described in the next subsection, add to the sensibility of the transition.

It may be easier to understand how RealityFlythrough works by envisioning the following concrete example. 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. RealityFlythrough operates in a much more forgiving environment: the virtual walls are not necessarily at right angles, they do not all have to be the same distance away from the viewer, and they do not have to have a uniform opacity.

For the applications we envision, the user will spend the majority of the time viewing real images generated by a live camera. A flythrough will consist of moving from camera to camera with transitions displayed in the intervening space to give the user cues of the spatial relationship between the cameras. During these transitions, there will likely be mismatched objects, ghosting, and tears, but these defects are unavoidable given the environment we want the system to work in. Without an accurate model of the geometry of the space, without accurate information about the location and position of the cameras, and given the live and real-time requirements of our system, it would be very difficult to avoid defects. We are careful to reveal these defects to the user rather than smoothing them over with blurring, because their very presence helps the user make sense of the transition. Fortunately, the defects do not appear to harm more than just the of the flythrough. It is only a secondary goal of the system to generate pleasing imagery. First and foremost, the system should allow a user to act, and in the scenarios we envision, action requires real-time access to the data. The benchmark for such a system is not photorealism [1], but rather what the system can enable.

RealityFlythrough works in the wild because there is little information the system requires about each camera, and no preprocessing is required to render the transitions. The position of the camera can be obtained from whatever locationing technology is desired (we use WAAS-enabled consumer GPS's). The lateral direction can be determined with a digital compass (we use the compass provided by the GPS), and the pitch and roll by an inclinometer (we have ignored these in this paper). In addition to the location information, we require the angle of view for each camera.

The angle of view is a constant that is determined in a calibration procedure for each camera [1] (or camera lens if dealing with cameras that have interchangeable lenses). Before each transition is started and while the transition is in progress, the most recent information about each camera is used in computing the transition.

2.1 Transitions

There are a number of enhancements that can be made to a transition to increase its sensibility.

Image overlaps. The first enhancement is to blend the images involved in a transition to reveal inconsistencies while still being pleasing to the eye. Most importantly, blending makes certain kinds of transitions (such as those that involve forward or backward motion) sensible. We found that the best blend is achieved by showing both the starting and terminal images at full opacity where there is no overlap, and doing a straight alpha blend from the starting image to the terminal image where there is overlap. This low-cost rendering technique has a "stitching" effect for side-side overlaps on sideways or rotational transitions, and a picture-on-picture zooming effect for forward transitions.

Image gaps. The real key to making transitions successful involves determining what images to display during a transition if the starting and terminal images do not overlap. We do not display the images from all cameras covering the current view because there are cases where too much information is presented to the user, resulting in confusion. Instead, a transition is limited to displaying at most two images simultaneously. A long transition is composed from a series of simple two-camera transitions. We determine the images that best fit along the path from starting camera A to terminal camera B using a fitness function, and then construct a series of transitions between each of these images while continuing along the path from A to B.

Additionally, to avoid the startling and confusing effect of having to make sense of too many images in a short amount of time, we developed the following three heuristics: (1) The current image should stay in view for as long as possible, (2) once the terminal image can be seen from the current position, no other transitions should be considered, and (3) there should be a minimum duration for sub-transitions to avoid the condition of images "flashing".

Projection-wall distance. One outstanding issue is how the distance from each camera to its virtual projection wall is calculated. A 3d geometry of the space could be pre-acquired [2], and a reasonably accurate rendering of the scene could be achieved by projecting the camera image onto the model, but a model is not likely to be available in the wild. Besides, often the more interesting subjects being viewed are dynamic (e.g., people), and cannot be modeled apriori. Without an accurate model of the world, we are forced to compromise by projecting the image onto a flat surface some "good" distance away, introducing distortions by pushing nearer objects and pulling farther objects to the plane of the screen. Unfortunately, what is good for one transition may not be good for another because in one transition a foreground object may wind up being dominant while in another a background object is dominant. We must choose a distance that generally looks good, and accept the imperfections in some of the transitions. Fortunately, we have not found the imperfections in the transitions to substantively affect sensibility.

For still photographs (see the next subsection), we have been manually calculating—estimating—the distance between the camera and the most dominant object in the image. For the telepresence experience that we present in Section 4, we simply pre-selected a distance that was close to the maximum distance between most places in the setting (30 meters). More generally, there are several possible approaches that deserve consideration (roughly by

[1] The calibration of a camera is done independently of the environment and of other cameras, so this does not affect our ability to function in the uncalibrated wild.

increasing difficulty and presumed accuracy): (1) a typical middle-distance is used that has proven to work well in practice, (2) the user controls the distances and can adjust them as deemed necessary, (3) cameras that have the ability to autofocus provide the range information, (4) on-the-fly image processing estimates distances, and (5) meta-information in the form of bounding polygons is used to help determine the geometry of a camera's cone and hence the distance to the wall.

2.2 Additional Sense-making Devices

Beyond transitions, we have included several other sense-making devices in the RealityFlythrough engine.

Image gaps. Typically, the number of live video cameras in a scene will be inadequate to create a seamless telepresence experience. Consequently, we boost our camera density by inserting still photographs into the environment, which then fill gaps between live video during a transition.[2] By giving the user some visual cue that they are looking at an old still image rather than a live video feed (by antiquing the image, for example), the system provides the user with additional context for how the live video feeds relate to one another and to the setting at large.[3]

Even when using still photographs in conjunction with live video, 100% camera coverage is unlikely. To give the user a sense of the amount of ground covered during a transition that has a gap between images, we add a virtual floor grid (inspired by ones used in old arcade games). The grid is partially visible in Fig 3. A properly scaled map, in perspective, might also be effective.

At low video camera densities, users also need guidance as to where live video is available in the setting, especially when the video cameras are moving around. We currently provide a birdseye view (see Fig.2b) that shows a map of the space (if one is available), the locations and directions of all cameras, and an indication of which cameras are streaming live video. The birdseye map shows a cone being emitted from the current camera indicating the approximate area of coverage of the current image.

Pacing of transitions. Lastly, the sense of depth and distance lost by a camera's flattening of 3d to 2d can be recaptured in part by adding the context of speed to transitions. By having transition speed be constant or user-controllable, the user can develop a feel for how long it takes to move a certain distance and use this as an additional cue for the spatial relationships among cameras.

2.3 Moving Video Cameras

In order to support live video, we make use of the H323 video conferencing standard. There are a number of video conferencing applications that implement this standard, and they run on a wide variety of platforms. In particular, we have adopted OpenH323 (http://www.openh323.org), an open source solution that runs on Windows, Linux, and the Pocket PC. We are currently using unmodified clients that call into a Multipoint Control Unit (MCU) that has been incorporated into the RealityFlythrough engine. The MCU is a modified version of OpenMCU (also available at http://www.openh323.org) that caches the most recent frame of each incoming video stream so that the RealityFlythrough engine always has immediate access to the latest images.

The position and orientation of each camera is captured by a separate client application and transmitted to the RealityFlythrough server using a SOAP/RPC interface. With the GPS devices we have experimented with so far, the

[2] We hope to soon replace the manually positioned still images with still images captured from the live video feeds thus allowing the system to work with no preconfiguration.

[3] We were surprised to find that a flythrough composed of a few still photographs and no video is both compelling and effective for applications like real-estate.

data is refreshed every two seconds.

3 Related Work

There have been several approaches to telepresence with each operating under a different set of assumptions. Telepresence [3], tele-existence [4], tele-reality [5], [6], virtual reality and tele-immersion [7] 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 ubiquitous form of virtual reality. Tele-immersion describes the ideal virtual reality experience; in its current form users are immersed in a CAVE with head and hand tracking devices.

RealityFlythrough contains elements of tele-reality 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 telepresence 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 spatial relationships between the views that are available.

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 otherwise 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 that is good enough for movie special effects ("The Matrix Revolutions" made ample use of this technology). An alternative tele-reality approach assumes apriori acquisition of a model of the space [2], 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 result is live, real-time exploration of the world, or command/control support in a disaster response scenario.

4 Telepresence Experience

Since RealityFlythrough is the first system of its kind to be built and used, we had only hunches of what may be its defining characteristics and issues. The design of the experience was not to measure the usefulness of RealityFlythrough, but rather to discover the issues that may lead to further improvements to the system. As is the goal of most qualitative studies, we wished to generate hypotheses [8]. What were the qualities of the system that required further study? As the first system of its kind, what would happen? Would the system work? How would people react to this new technology? What would the comfort level be? What are the shortcomings of the technology?

To answer these questions, we assembled an end-to-end telepresence system using our RealityFlythrough engine, free H323 video-conferencing software, and commodity hardware. We designed a simple experiment to mimic what we feel would be a common usage scenario of telepresence. Two subjects, working together, were asked to do a flythrough of our main campus food court (and social center), called the Price Center, to determine if there was any reason to physically go there.

4.1 Experimental Set-up

Technology. For positioned and oriented video capture, we assembled a video camera package that consisted of an inexpensive logitech web camera ($50) duct-taped to a WAAS-enabled Garmin eTrex Vista GPS that has a built in digital compass ($300). The camera package was connected to an 802.11b-equipped laptop. Although bulky and missing some features, this setup provided most of the information we needed for a short-term study. We were missing pitch and roll data (i.e., how the camera was tilting), but this was managed by asking the camera operators to hold the cameras as level as possible. This simplification is not a concern for our questions, because pitch and roll will ultimately be the most accurately calculated values. The available instruments that detect pitch and roll are relatively insensitive to external perturbations when compared to GPS's sensitivity to cloud cover or a magnetic compass's sensitivity to the proximity of metallic objects.

The flythrough operators were in our laboratory using a Dell Precision 450 running Windows XP with a 128MB nVidia QuadroFX 1000 graphics card, and a 19 inch Dell LCD display (1280x1024). For simplicity, the RealityFlythrough engine ran directly on this machine.

Camera operators at food court. For this experiment, we had three camera operators, each equipped with the camera package. They were asked to hold the cameras level and to try to behave as naturally as possible given the equipment they were carrying.

Flythrough operators. A key choice made in the set-up of this study was to use a *pair* of operators (a pilot and co-pilot, if you will), rather than a single operator. The primary reason for this choice is that the collaboration induces the operators to *naturally* verbalize for the purpose of completing the task. This approach to experimental observation, known as *constructive interaction* [9, 10], allows us to get access to the operators' thoughts through their substantive interactions without prompting on our part, which could disturb the activity or otherwise bias the study. A secondary consequence is that the two operators together might be more effective than a single operator, because the co-pilot may see something the pilot has overlooked (as one example). However, we note both that paired work practices are common (and hence our set-up is realistic in this sense), and our questions are qualitative in nature and not tied to single-operator use.

We selected two garrulous colleagues to perform the flythrough as a team. They were encouraged to chat as they tried to answer the following questions: Do you see anyone you know? What is the person doing? What inferences can you make about his/her availability? What is the weather like? Are there any events going on? Is the Price Center too crowded? Too empty? Are the lines too long? Is the bookstore open?

The specific questions we wished to answer were: Does RealityFlythrough help the user solve the tasks? Could the tasks have been solved using simpler technology? (e.g., by just looking at three video feeds on a monitor.) Did the transitions add value? If there was no added value, what was the reason? Would there be added value if there were more live cameras? How does the user conceptualize the system? Is it seen as a seamless system or a series of disjoint images? Did the imprecise location information cause confusion? Did the transitions cause confusion? Did RealityFlythrough promote exploration? Can we imagine using this tool to take a walk? To be a virtual tourist? To do virtual window shopping?

Actors at foodcourt. We planted three actors whom the subjects either already knew or were introduced to prior to the start of the experiment. The actors were given specific tasks to perform while at the food court, but the subjects did not know this.

4.2 Observations

The telepresence experience worked end to end for 30 minutes. The server crashed for an unknown reason at the 14 minute mark, but it was quickly restarted. Throughout the experiment we recognized that the position and orientation of the cameras as displayed on the birdseye view corresponded with the images that the video cameras were displaying. The only complaints we heard about inconsistencies with the camera positions occurred just before the server crash when the server was quickly degrading.

The frame rate of the incoming video feeds appeared to be about two frames per second. One subject reported he was getting a "headache from the jitter." The low frame rate and the side-to-side motion of the camera created "a sense of walking. Like a penguin." The quality of the video feeds was good enough that movement, clothing, and colors ("the red helped to cue me") could be recognized, faces were recognizable within about three meters of the cameras (See Fig. 3). "People I don't know are just a blob; people that I happen to know what they're wearing that day I can recognize. I don't think I would have known that was John unless he was wearing the big jacket." Distinguishing features did stand out: "I haven't seen that silly little baby in some time. They should stand out because of their size." And later, "Small child! But not our small child."

There were many instances when the video quality was good enough to recognize behavior: "Is that John on the steps? Yeah, that's John. It looks like he's eating something or reading something." The behavior of others was recognized, as well: "There's obvious proof that the Sunshine Store is open. People going in." And when activity on stage was spotted, "All right! We got a dance troop. Very cool." An individual was seen sitting at table on a telephone: "On the cell phone; she's going to be there for awhile."

For the first half of the experiment, many of the transitions between the cameras did not display the filler photographs that were designed to provide additional context. The grid lines were present, however. There were also sudden jumps in the orientation of the camera just as the transition ended. "Nothing like an about face. I wanted to look at the stairs. Now I can't." Using a cell phone, we instructed the camera operators to reduce the frequency of camera panning and to pan more quickly. Once the message reached all three operators, the filler photographs appeared as expected, and the sudden jumps at the end of transitions were much less severe. The improved transitions prompted the comment: "Those images backfilling like that really does help you when moving between images." We will explain why reduced panning helped the transitions in Section 4.3.

The subjects had complete control over which cameras were used during the flythrough, but no control over the cameras themselves. They spent much of the time hitchhiking on individual cameras rather than moving between cameras. In many cases, there were specific reasons voiced for changing cameras: "See if we can get a close-up." "Ok, let's go to the bookstore." "Let's turn to the right." And, "Let's see, can we make a minor adjustment here."

4.3 Analysis

The experience was a success. Our primary question was whether or not we could create a telepresence experience in the wild, and the answer is unequivocally yes. There is plenty of room for improvement, but by and large RealityFlythrough exceeded our expectations.

Position accuracy. Perhaps the biggest surprise was that GPS location accuracy proved to be good enough for RealityFlythrough. It is hard to tell how accurate the GPS data was, but the view in the video feeds matched well with the position and orientation of the cameras as displayed on the birdseye view map. Our camera operators noted that the GPS devices were displaying an error of roughly nine meters. After analyzing the transitions between cameras, we realized that RealityFlythrough is quite tolerant to location inaccuracies. Transitions remain sensible as long as there is some image overlap. With the wide field of view obtained from our cameras in an outdoor setting, location errors would have to be very large to make neighboring images not overlap. Much more important, it turned out, was the accuracy of the orientation data because the angle of view of our cameras was only 45 degrees. Fortunately, the error of our digital compasses was very small.

Paradoxically, the crash of the system provided the clearest indication that the position information was helpful to the subjects. Right before the server crashed, the video feeds were still being transmitted, but the position and orientation of the cameras were no longer being updated. This failure was immediately noticed: "Ok, let's go to the bookstore. Whoops. I guess that wasn't the bookstore." This prompted discussion between the two subjects as they tried to resolve why there was not a correspondence between the images and the positions as indicated on the map. Other than this episode, the subjects made no mention of transitions being problematic. Once the server was back on line, the subjects spent a few moments confirming that the correspondence was back, and then continued with the experiment. One noted: "That is an interesting habit; wanting to confirm that it [the camera's view] is looking where this [the arrow on the birdseye view] is... When I get disoriented looking at the photo, I look at the image on the right hand side to reorient myself."

Video quality. Early in the experiment, the poor quality of the video feeds seemed to be affecting the user experience. Before the subjects had managed to locate someone they knew, they were pessimistic about their ability to do so. As they learned to accept the limitations of the tool given to them, and as they became more adept at inferring behavior from incomplete information, they made fewer comments about the image quality. During post-experiment questioning when discussing the quality of the video and possible areas of improvement, one subject commented: "I think we had an idea of who would be there for awhile. We didn't need the close-up expression for that. The body movement, the reading the paper, taking off the jacket, we saw they were going to stay for awhile. There were a lot of motions for which large body motions told the story."

Yet, improving the video quality would certainly improve the user experience and increase the utility of RealityFlythrough. Watching a sporting event at two frames per second would not be tolerable.

Transition flaws. The problems we encountered with missing filler photographs in the first half of the experiment had two causes. The first is related to the amount the terminal camera's orientation and position changes between the start and end of a transition. At the start of a transition, a path between the starting and terminal cameras is calculated and the images that best fit along that path are selected. If, after the transition has been calculated, the terminal camera changes its orientation drastically, the filler images selected may no longer be relevant to the transition. To solve this problem we need to modify the algorithm that computes transitions so that it selects filler images just-in-time.

The second cause had to do with the slow update of the position information. The Garmin GPS devices only provide updates every two seconds. This frequency is adequate for position information, but not for orientation. During a transition, the orientation of a camera would suddenly jump by a large amount, and the corresponding image would either do a sudden shift on the screen or disappear altogether if the user's view no longer corresponded to the camera's view. Once the transition was finished, the user would be hitchhiking on the terminal camera, and the camera's image would jump back into view. Despite these flawed transitions, it is surprising that the subjects were still able to make sense of the movement as evidenced by this comment regarding one of the poor transitions: "Nothing like an about face. I wanted to look at the stairs. Now I can't." The solution to this problem is to use different hardware to get more frequent orientation updates.

Even without the filler photographs, the transitions were still sensible to the subjects, apparently because the camera motion and the floor grid conveyed the necessary spatial information. The tolerance for a low density of imagery validates one of our design goals of graceful system degradation.

Conceptualization of experience. One of our research questions is how the user conceptualizes the RealityFlythrough experience. Is it conceptualized as a seamless experience or a series of disjoint images? The language that was used by the subjects during the experiment can provide some clues. Statements like "Let's go to the bookstore." and "Let's turn to the right." indicate that the user has a sense of spatial navigation, even a bit of "being there".

In the post-experiment interview, the word "wander" was used to describe their use of the system: "I think the ability to wander through the space, to use all cameras as if they are one, is definitely important." And, "...I really didn't need to intentionally go wandering around" (because the camera operator moves). The use of the wandering metaphor suggests that the experience was spatial, continuous, and unconstrained in nature, rather than, say, browsing a series of disjoint images. One subject confirmed this: "If the images were separated, it would have been more difficult to have gotten the same sense. I've occasionally seen these surveillance cameras where they have four images. You're there but you're not there; it's very distracting."

Hitchhiking metaphor. Early in the conception of RealityFlythrough, *hitchhiking* was considered to be one of the key metaphors we wanted to enable. We envisioned users hitching a ride on a camera and then jumping to another camera when the original camera's view no longer suited them. What we discovered with this experiment is that the hitchhiking metaphor dominates when the task is exploration and the cameras are moving considerably. As noted earlier, the subjects spent much of the time sitting on a single camera and switched between cameras primarily when a different angle or position was desired. When asked about this one of the subjects admitted, "Except when I intentionally wanted to change to the other side, it was more comfortable just to stay on a single camera for the duration. There was enough actual movement in the person who had the camera that I really didn't need to intentionally go wandering around. If they had been static cameras posted on a light post or something, and we were integrating the images, then I would probably wander around more."

Given the tasks our subjects were asked to complete, riding a single camera feed was an effective solution. We can imagine other tasks, though, where this would not be the case (e.g., monitoring the entrance to the bookstore to see who enters and leaves). We plan to support this by enabling what we call the Virtual Camera metaphor. See Section 5.

Remaining questions. Does RealityFlythrough help the user solve the tasks? Yes. The subjects reported that they had a pretty good idea of who was there, who would be around for awhile, what events were going on, what was open, and what the weather was like. The camera operators confirmed that the subjects' inferences were correct. As our set-up could be easily improved in terms of frame rate and image quality, we surmise that utility can only improve over time.

Could the tasks have been solved using simpler technology? The tasks probably could have been solved using simpler technology. However, our subjects noted that having the single view on the world helped with sensibility.

They did wonder if they could have had a similar experience by using three video cameras that had fixed locations. However, this depends upon spaces being pre-instrumented with video cameras and made available to others, which is not a given. We are embracing mobile cameras because it is apparent they are becoming ubiquitous. If we can handle mobile cameras we can certainly handle fixed cameras.

Did the transitions add value? The transitions seemed to help the subjects make sense of how video streams related to one another. “Those images backfilling like that really does help you when moving between images.” The only time confusion was voiced was when the system was crashing and the camera positions did not correspond to reality. Even when the background filler images were not being displayed properly in the first half of the experiment, the transition motion, and the grid lines appeared to be sufficient for conveying spatial information.

Did RealityFlythrough promote exploration? Comments like “Let’s go to the bookstore.”, “Let’s turn to the right.”, and “See if we can get a close-up.” indicate that the subjects were exploring.

Can we imagine using this tool to take a walk? To be a virtual tourist? To do virtual window shopping? Given the current quality of the video feeds, it is a little difficult to imagine using this system in place of taking a walk (assuming taking a physical walk is possible or desirable). Virtual window shopping would not be possible with our current set-up unless the cameras were close to the merchandise. Virtual tourism is certainly possible. We can imagine many university alumni would be interested in a nostalgic virtual visit to the Price Center.

5 Conclusion and Future Work

We are rapidly moving toward a world where personal networked video cameras are ubiquitous. We have demonstrated that telepresence with such cameras can be made to work in the wild.

Using commodity hardware, and standard video conferencing software, we were able to create a live, immersive, and compelling telepresence experience. The wild is far removed from the comfortable laboratory. Camera quality is diverse, locationing technology is imprecise, and equipment capabilities are variable. The environment is dynamic; people and objects are constantly on the move. Lighting is inconsistent, and cameras move, tilt and sway with their operators. By employing a number of sense-making techniques to help a user understand the spatial relationships between images, we have made telepresence work even in this hostile environment. The only inputs to the system required are: an image (or stream of images), and the position from which the image was captured. The higher the fidelity of these inputs, the better the user experience will be, but we hope to provide some user experience even at the low end of this spectrum.

We constructed a telepresence experience in the wild, and from this study we are able to draw the following conclusions: (1) Telepresence can be made to work in the wild. Although we did make some concessions in our experiment due to limitations in current technology, it is not difficult to imagine extending our results to any place in the wild. We were somewhat constrained by the deficiencies of GPS for determining the location of our sensors, but today’s heightened focus on location-aware technology suggests better solutions will soon emerge. We were also constrained to an environment that supported 802.11 wireless, but technology trends indicate that these environments will soon be ubiquitous. (2) The use of transitions to convey spatial information about the scene was well received by our subjects. Even though there were only three live cameras at the scene, our subjects still felt that the immersive feel of the first-person view was substantively better than the experience they may have had viewing an array of separate video feeds. (3) RealityFlythrough is tolerant to position inaccuracies. Even with reported GPS inaccuracies of roughly nine meters, the transitions were still sensible. (4) In contrast, orientation accuracy is critical, and in fact orientation updates should be as close to real-time as possible. They certainly need to be more frequent than once per two seconds, the rate that we obtained. (5) Some form of wide-area spatial abstraction that preferably does not have the distractive elements of a separate birdseye view is desirable. Our subjects consulted frequently with the birdseye view to get a big picture overview of the scene, but it was difficult for them to constantly make the cognitive leap between the 3d first person view and the 2d birdseye view. And, (6) the poor quality and jerkiness of the video detracted from the experience. There are numerous solutions to this ranging from improved cameras to anti-jitter software. The frame rate can be improved by using a more efficient video codec and by employing a flow-control protocol to throttle the video cameras that are not being viewed.

These conclusions and other insights from our experience with RealityFlythrough point to several possible lines of future work.

Improvements to transitions. Improving the sensemaking qualities of transitions is a high priority because of their importance in conveying contextual information. There are several promising approaches to improving them. (1) Use just-in-time techniques for path planning and camera selection to maximize the chance that the most suitable image is displayed. (2) Explore whether vision techniques such as "structure from motion" might be able to create a minimal geometric model to help with the projection of the camera image. And, (3) possibly use pre-processed stitching (mosaicing) techniques [5] on still images to help improve the aesthetics of some side-side transitions.

Virtual camera metaphor. The hitchhiking metaphor has dominated our design up to this point. Another compelling modality for RealityFlythrough is best described as the virtual camera metaphor. Instead of selecting the video stream to view, the user chooses the position in space that they wish to view, and the best available image for that location and orientation is displayed. "Best" can either refer to the quality of the fit or the recency of the image.

Incorporate birdseye view in main view. Although the "map was definitely useful," having it be the sole tool for navigation is not ideal because forming the continual correlations between the main eye-level scene and the 2d birdseye representation takes cognitive resources away from the flythrough scene and its transitions. We hope to be able to integrate most of the information that is present in the birdseye view into the main display. Techniques akin to Halos [11] may be of help.

Sound. Sound is a great medium for providing context, and could be an inexpensive complement to video. By capturing the sound recorded by all nearby cameras, and projecting it into the appropriate speakers at the appropriate volumes to preserve spatial context, a user's sense of what is going on around the currently viewed camera should be enhanced.

Control of cameras. The experiment subjects expressed the desire to have more control over the cameras, which is a move in the direction of telepresence. One way to accomplish this is to allow them to control the zoom, focus, exposure, and possibly orientation of the cameras (several more expensive cameras support this). Another is to provide a communication back-channel to the camera operators themselves.

Adaptability to hardware. The hardware used by the camera operators had ergonomic, aesthetic, and technical deficiencies. We hope to eventually use head-mounted cameras complete with integrated GPS, compass, and inclinometers, but we aim to support whatever hardware can provide an image and adjust the user experience accordingly.

References

[1] Hollan, J., Stornetta, S.: Beyond being there. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (1992) 119–125

[2] Neumann, U., You, S., Hu, J., Jiang, B., Lee, J.: Augmented virtual environments (ave) for visualization of dynamic imagery. http://imsc.usc.edu/research/project/virtcamp/ave.pdf (undated)

[3] Kuzuoka, H., Ishimo, G., Nishimura, Y., Suzuki, R., Kondo, K.: Can the gesturecam be a surrogate? In: ECSCW. (1995) 179–

[4] Tachi, S.: Real-time remote robotics - toward networked telexistence. In: IEEE Computer Graphics and Applications. (1998) 6–9

[5] Szeliski, R.: Image mosaicing for tele-reality applications. In: WACV94. (1994) 44–53

[6] Kanade, T., Rander, P., Vedula, S., Saito, H.: Virtualized reality: digitizing a 3d time varying event as is and in real time (1999)

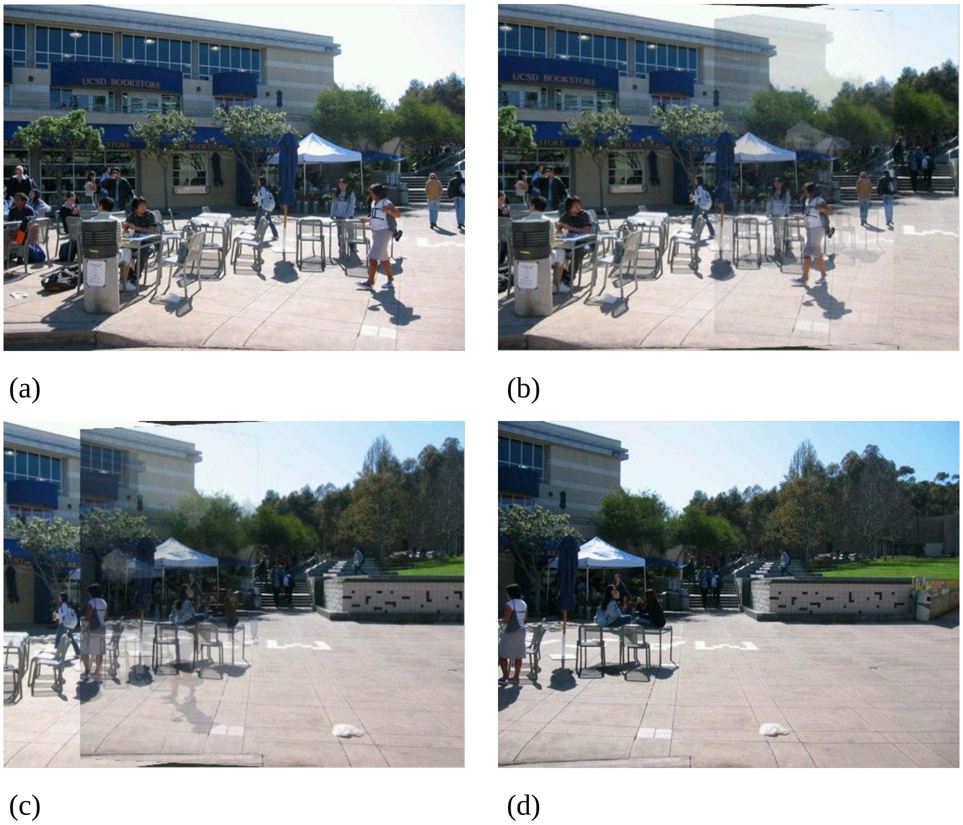
[7] Leigh, J., Johnson, A.E., DeFanti, T.A., Brown, M.D.: A review of tele-immersive applications in the CAVE research network. In: VR. (1999) 180–

[8] Strauss, A.L.: Qualitative Analysis for Social Scientists. Cambridge University Press, Cambridge (1987)

[9] Miyake, N.: Constructive interaction and the iterative process of understanding. Cognitive Science **10** (1986) 151–177

[10] Wildman, D.: Getting the most from paired-user testing. ACM Interactions **2** (1995) 21–27

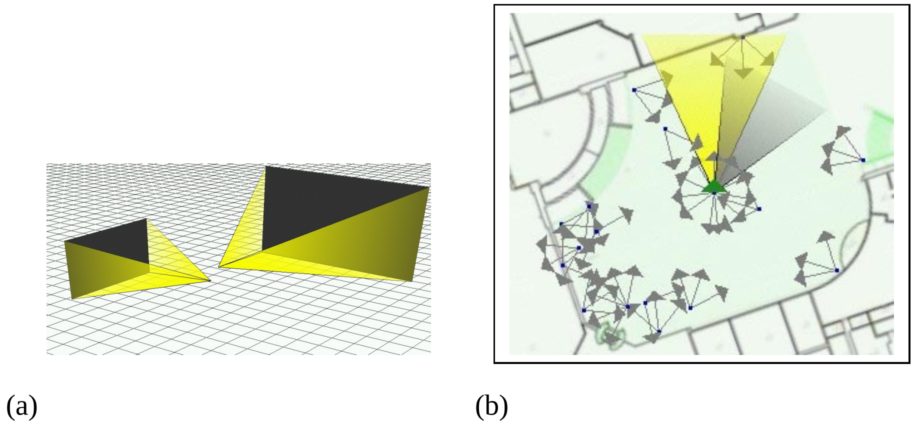
[11] Baudisch, P., Rosenholtz, R.: Halo: a technique for visualizing off-screen objects. In: Proceedings of the conference on Human factors in computing systems, ACM Press (2003) 481–488



(a) (b)

(c) (d)

Figure 1: A transition from image (a) to image (d). Images (b) and (c) show the transition in progress as image (a) moves off the screen to the left and image (d) moves in from the right. This transition corresponds to a rotation to the right. Note: misalignment exaggerated for presentation purposes.



(a) (b)

Figure 2: (a) An illustration of how the virtual cameras project their images onto a wall. (b) The birdseye view. The arrows represent the camera locations and directions of view. This picture corresponds to the transition in Fig. 1.

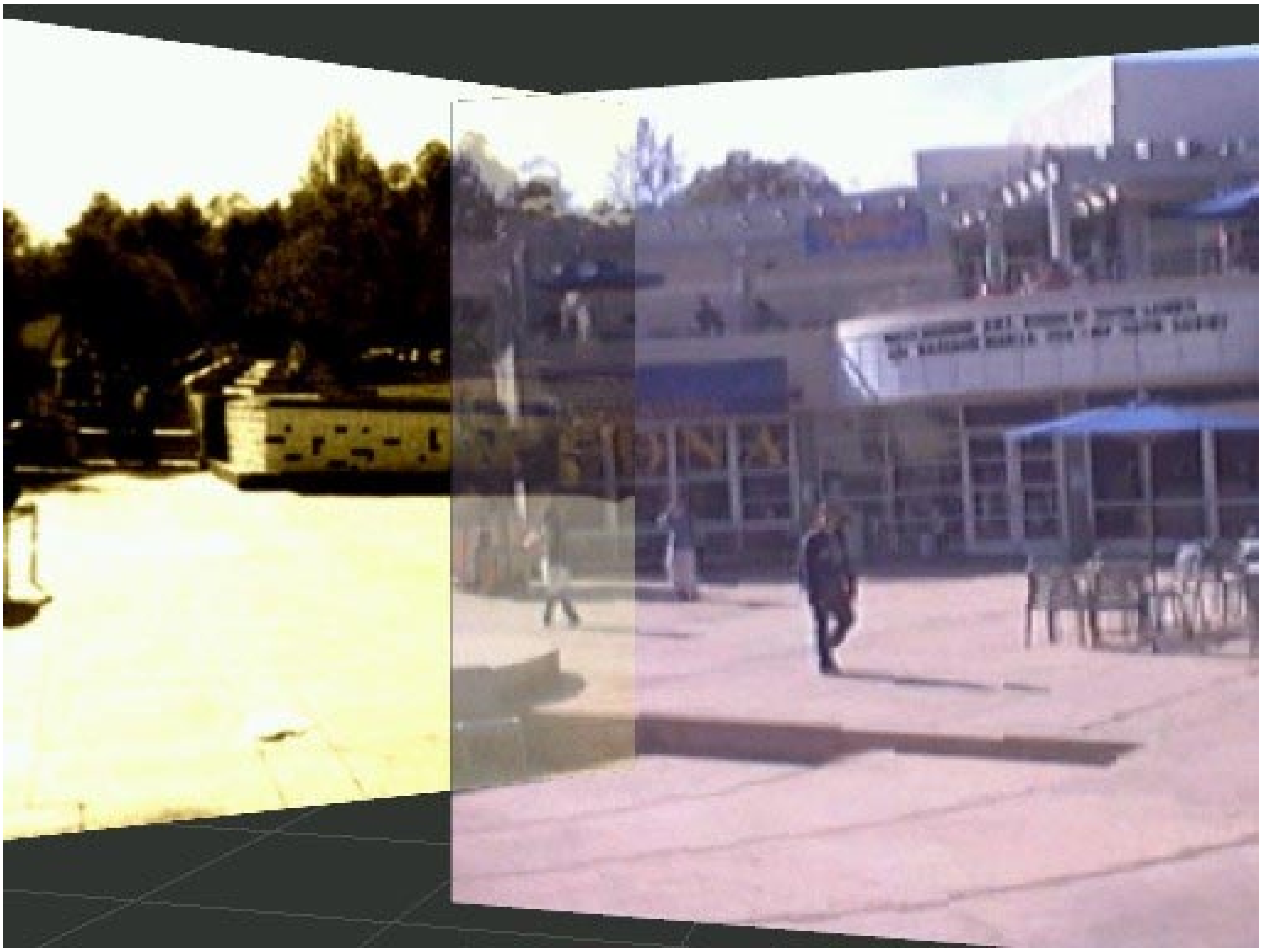


Figure 3: A transition in progress involving an antiqued filler image (on the left) and a live video feed (on the right).