Tele-Reality for the Rest of Us

Neil J. McCurdy, William G. Griswold
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114
{nemccurd,wgg}@cs.ucsd.edu

ABSTRACT
We are rapidly moving toward a world where personal networked video cameras will be ubiquitous. Already, camera-equipped cell phones are becoming commonplace. Imagine being able to tap into all of these real-time video feeds to remotely explore the world live. We introduce RealityFlythrough, a tele-reality/telepresence system that will make this vision possible. By situating live 2d images 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 images spatially relate to one another. By providing spatial cues in the form of dynamic transitions, we can approximate tele-reality and harness cameras in the wild. This paper focuses on the sensibility of these imperfect dynamic transitions from camera to camera. We present early experimental results that suggest that imperfect transitions are more sensible, and provide a more pleasant user experience than no transitions at all.

Author Keywords
Tele-reality, Telepresence, Image mosaicing, Ubiquitous computing

ACM Classification Keywords
Tele-Reality, Ubiquitous computing.

INTRODUCTION
We are rapidly moving toward a world where personal networked video cameras will be ubiquitous. Already, camera-equipped cell phones are becoming commonplace. Imagine being able to tap into all of these real-time video feeds to remotely explore the world in the present. We introduce RealityFlythrough, a tele-reality/telepresence system that will make this vision possible.

This work is supported in part by an HP University Mobile Technology Solutions gift, Microsoft Research, and the California Institute for Telecommunications and Information Technology (Cal-(IT)2).

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 cameras encountering the chaos of a disaster site. As they fan out through the site, they continuously broadcast their location and what they see to a RealityFlythrough server. The responders' central command could virtually explore the site by viewing these video feeds to get a sense of the big picture. Medics could locate the injured, firefighters could see potential flare-ups, and engineers could see 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.

There are many other applications for RealityFlythrough, ranging from allowing the disabled to remotely explore the world, to allowing sports fans to remotely fly around a stadium selecting the optimal vantage point for viewing the game.

An early description of tele-reality in the academic literature was presented by Szeliski [5]. He suggests that the ultimate in tele-reality is dynamic tele-reality, a live immersive real-time flythrough of the world. The distinction between tele-reality and telepresence is subtle and not necessarily enforced. Telepresence typically involves the remote control of a robotic camera [4], while tele-reality builds a model by using multiple cameras and allows what are called novel views from locations that are not covered by cameras. Much research has been done by the graphics and vision communities in texturing virtual reality with photos, with a focus on creating photorealism at every point in space [3]. These systems require extensive preprocessing of the images and special cameras, rigs, scaffolding, and lighting to achieve the effect. These systems are solving a different set of problems and are using a different set of assumptions, and will not work in the wild, where cameras are moving, and the images are live video feeds that cannot be preprocessed.

RealityFlythrough addresses these problems by relaxing the requirement for photorealism [2] during the transitions be-

Figure 1. A snapshot of a transition in progress.

tween images. Transitions are a dynamic 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 model of the space. Although it is possible to stop mid-transition to see a novel view (as shown in Figure 1), the emphasis is on displaying real camera-generated images. The transitions from camera to camera are mainly provided to help the user make sense of how the images are related to one another spatially.

Crucial to the success of this work is the sense-making qualities of the transitions and the comfort the user has with viewing them. It is for this reason that we focus on studying the *transition* in this paper. In the next section we will discuss how the transitions in RealityFlythrough are achieved, and in the subsequent section we analyze the effectiveness of the transitions.

APPROACH

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 dynamic transitions displayed in the intervening space to give the user cues of the spatial relationship between the cameras. There will likely be mismatched objects, ghosting, and tears during the transitions, but these defects are unavoidable given the environment we want the system to work in. 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.

RealityFlythrough will work in the wild because there is very little information the system needs about each camera. The position of the camera can be obtained from current locationing technology. The lateral direction can be determined with a digital compass, and the vertical direction by an inclinometer. In addition to the location information, we require the field of view for each camera. The field of view is a constant that is determined in a calibration procedure for each camera (or camera lens if dealing with cameras that have interchangeable lenses). None of the required information needs to be known in advance, as no preprocessing is necessary. Before each transition is started the most recent information about each camera is used to calculate the best transition in real-time.

At this stage of the research, a key question is how effective our approach to transitions is. To focus on the qualities of transitions and dramatically simplify their assessment we have explored this question through the use of stationary still images in a space, rather than live video. Our early results with video indicate that video adds, rather than detracts from the sensibility of the transitions and certainly contributes to the immersive feel of the experience.

A benefit of doing this experiment with still images is that we found that the immersive feel achieved with just a few photographs of a space suggests that we can use still photographs to increase our camera density. By giving the user some visual cue that they are looking at an old still image rather than a live video feed (for example, making the image black and white), we can provide the user with additional context for how the live video feeds relate to one another. It may also be possible to more accurately measure the position of the still images, and possibly use preprocessing techniques to improve the quality of the transitions and give the user a better sense of the geometry of the space [3].

IMPLEMENTATION

RealityFlythrough works by situating 2d images in 3d space. For each camera, a virtual camera is placed at a corresponding position in virtual space and oriented in the same direction as the real camera. The camera's image is then projected onto a virtual wall. Computing the distance between the camera and the wall is problematic and will be addressed shortly. 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. A transition between camera A and camera B 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. We use OpenGL's standard perspective projection matrix when rendering the images used during the transition. 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.

The above approach is adequate for only very simple transitions, but there are a number of improvements that can be made to increase sensibility. The first improvement is to introduce blending of the images to reveal inconsistencies while still being pleasing to the eye, and most importantly to make 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 *to* and *from* images at full opacity where there is no overlap, and doing a straight alpha blend from the *from* image to the *to* image

where there is overlap.

The most important improvement, and the key to making transitions successful, involves determining what images to display during a transition. We do not display the images from all cameras covering the current view because this is potentially confusing because it pushes too much information at the user. Instead, we only ever display two images simultaneously. We build a transition by composing it out of a series of simple two camera transitions. We determine the images that best fit along the path from camera A to camera B using a fitness function, and then construct a series of transitions between each of these images while continuing along our path from A to B.

The above approach alone does not result in consistently sensible transitions because time is required by the visual system to process inconsistencies. To avoid the startling 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 *to* image can be seen from the current position, no other transitions should be considered, and 3) there should be a minimum duration for subtransitions to avoid jumpiness.

We will just briefly mention the other sense-increasing artifacts that are in our current RealityFlythrough engine. Since there will never be 100% camera coverage, we added a virtual floor grid (inspired by ones used in old arcade games) to help give the user a sense of the amount of ground that was covered when no images are present. Also, to help with inadequate camera coverage, there is a birdseye map that shows a map of the space (if one is available) and the locations and directions of all cameras. In addition, on the birdseye map, a cone is emitted from the current camera indicating the approximate area coverage of the current image. The cones of other cameras can be viewed by mousing over them. Navigation in the flythrough can be performed by either clicking on cameras in the birdseye view or using keyboard controls similar to those found in current immersive video games. We have found that the user's intentions in the navigation informs their sense-making. We use the speed of the motion as an additional cue to indicate the distance traveled. This is valuable, because one of the problems with 2d images is that they do not contain complete spatial information. The sense of depth and distance is lost unless put into context.

We now address the problem of calculating the distance between the camera and the virtual wall upon which the camera's image is projected. We have been manually calculating this value as the distance between the camera and the most dominant object in the image, but this will obviously not work in the wild. Cameras that have autofocus already calculate this value; it would be nice if there was an interface for obtaining it. In the meantime we can either use a separate range finder to help us with the calculations, or the distance could be inferred by knowing something about the geometry of the space. Another possibility is that some typical middle-distance value will work will in practice, given our transition approach to sense-making. We are also considering incorporating meta-information in the form of bounding polygons to help prevent images from bleeding through walls, so this same meta-information could be used to determine the geometry of the camera's cone and hence the distance to the wall.

EXPERIMENT AND RESULTS

In order to learn more about the effectiveness of transitions, we created another version of RealityFlytrough that was identical to the original except that no transitions were performed when a user switched between cameras. An initial pilot study demonstrated the difficulty of obtaining conclusive quantitative results, due to the large number of experimental variables that must be controlled. There is a high variability in the experience and abilities of the users that would make statistical comparison difficult. The experiment that resulted was designed to help us answer the following questions: 1. How is the user's behavior affected by the transitions? 2. Do transitions help the user more quickly grasp the spatial relationship between images? 3. Do users automatically understand transitions, or is this a skill that needs to be learned? and, 4. What adjustments to the transitions can be made to increase sensibility?

As users of the system ourselves, and by studying the results of the pilot study, we choose the following partial operationalization of user behaviors for our questions: For question one, do users who do not have transitions flip back and forth between images more often (presumably trying to figure out how the two images relate)? For question two, do users who do not have transitions linger longer in certain parts of the space, trying to make sense of how the images relate to each other? For question three, do users who have transitions show or voice confusion during certain transitions? Question four would be answered through general observation and the results of a post-experiment questionnaire.

The experiment we constructed was designed to give the users a very concrete task to provide us with results that could be compared across all users. Each user was randomly given one of the two versions of the system and was given two minutes to remotely explore a portion of the ground floor of a 1500 square foot house. 31 images were made available that gave nearly complete coverage of three rooms. After exploring the space, the subject drew a floorplan from memory and tried to position as many objects as he/she could recall on the plan. The subject was not allowed to consult the images while doing the sketch, but was given a list of objects that may have been present to help with recall. During the exploration, the users were allowed to use the birdseye view to glean information about the relative positions of the cameras, but the birdseye view did not contain a map of the house.

Eleven subjects participated in the experiment. Six saw transitions between images, and five did not. We will identify the former subjects as the transitions group and the latter as the no-transitions group. Analysis of the resulting floorplan sketches is subjective and inconclusive. More experiments need to be done to control for the experience subjects bring to the task. First-person shooter game experience, innate spatial ability, comfort with spatial abstractions, and comfort with computers all played a role.

All participants were given the chance to use both versions of the experiment at the conclusion of the study, and there was unanimous agreement that transitions are better than no transitions. There were two exceptions to this sentiment during the pilot study, one of which may have been prompted by the way the pilot study was set up. The other case cannot be attributed to flaws in the experiment, and appears to be a genuine preference for no transitions. This subject was exhibiting all of the signs that indicate he was doing transitions mentally in his head (repeatedly flipping back and forth between two images), so it is interesting that he preferred not to have them. However, he said that he did not like the time required for a transition to take place, suggesting a desire for speed or efficiency.

To answer questions one and two, we present the following results: Two of the five no-transitions subjects spent about half their allotted two minutes stuck in the hallway which was covered by less than one quarter of all the images. No transitions subjects exhibited this behavior. A third no-transitions subject who is a hardcore gamer spent a little extra time in the hallway and did a fair amount of flipping back and forth between those images. A fourth no-transitions subject did not linger in the hallway, but was slow and methodical and only got to 3/4 of the images before time expired. This contrasts with the transitions subjects, all of whom covered the space completely and saw all images. For the subjects who did linger in the hallway, no extra detail about the hallway was revealed in their sketches of the floorplan. We should mention that the fifth no-transitions subject had what was clearly the worst floorplan sketch and apparently had no concept of the space being explored, but it's hard to tell why, so we shall ignore the practices used by that subject.

What the above results indicate is that subjects who did not have transitions had more difficulty making sense of the images they saw and had to move more slowly through the space or do more flipping back and forth between images to compensate. Subjects who had transitions may or may not have had more comprehension of the space, but it is clear that they thought they understood it because they did not linger.

We now address question three: There were several instances where transitions subjects showed surprise or confusion during their explorations, even though they had transitions to help them. These cases fall into two categories. The first category involves walking through walls, and the second, poorly 180 degree turn transitions that turned towards a wall, rather than away from it. The ability to walk through walls is a useful feature we want to include [2], but it was clear from these experiments that a feedback mechanism needs to be employed to alert the user of this odd phenomenon.

We received a comment from one subject that speaks to the naturalness of the transitions. He said that the rotations were more natural than the backward and forward translations, and that the latter took some getting used to. This is consistent with our experience. As expert users now, we are quite adept at internalizing the myriad sense-making cues, and while the transitions cannot be described as natural, they do seem to convey the information required for sense-making. Browser style *Back* and *Forward* controls would be useful to help the user see a transition multiple times if there is confusion. Repetition is a good sense-making device.

It is also interesting to note that the user interface requires some getting used to. We saw a number of the transitions subjects having difficulty with the keyboard interface, and many of them resorted to using the less natural birdseye view. Using the birdseye view requires constant translation between the 2d and 3d worlds, which detracts from the user experience. Ideally we would like to be able to convey all of the information that is present in the birdseye view on the main screen using techniques similar to those described in [1]. We received several comments about how much easier it would have been to navigate if the subjects had been able to move around the space as freely as we did in the post-experiment demos. The user interface clearly needs some work to make it more suitable for novice users.

CONCLUSION

We have described a tele-reality system that will work in the wild. It employs several sense-making techniques to help the user make sense of the spatial relationship between the images that are captured from adhoc cameras whose locations are dynamic and imperfectly determined. We have shown that the dynamic transitions we use to convey this information are more sensible than no transitions at all.

REFERENCES

1. Baudisch, P., and Rosenholtz, R. Halo: a technique for visualizing off-screen objects. *Proceedings of the conference on Human factors in computing systems*. ACM Press (2003), 481–488.
2. Hollan, J., and Stornetta, S. Beyond being there. *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press (1992), 119–125.
3. Kanade, T., Rander, P., Vedula, S., and Saito, H. Virtualized reality: digitizing a 3d time varying event as is and in real time, 1999.
4. Kuzuoka, H., Ishimo, G., Nishimura, Y., Suzuki, R., and Kondo, K. Can the gesturecam be a surrogate? *ECSCW*. 179–.
5. Szeliski, R. Image mosaicing for tele-reality applications. *WACV94*. 44–53.