Pseudo-3D Video Conferencing with a Generic Webcam

Chris Harrison Scott E. Hudson
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{chris.harrison, scott.hudson}@cs.cmu.edu

Figure 1. The video conference window acts like a virtual portal into the remote participant's space. As the viewer moves their head, the perspective of the remote environment changes. Motion parallax provides a 3D illusion. Note how the objects in the background, like the decorative vase, screen, and table move relative to the remote video conference participant.

Abstract

When conversing with someone via video conference, you are provided with a virtual window into their space. However, this currently remains both flat and fixed, limiting its immersiveness. Previous research efforts have explored the use of 3D in telecommunication, and show that the additional realism can enrich the video conference experience. However, existing systems require complex sensor and cameras setups that make them infeasible for widespread adoption. We present a method for producing a pseudo-3D experience using only a single generic webcam at each end. This means nearly any computer currently able to video conference can use our technique, making it readily adoptable. Although using comparatively simple techniques, the 3D result is convincing.

1. Introduction

Conversing with someone via video conference provides a virtual portal into their space. However, the video stream is both flat and fixed. This limits the realism of the face-to-face experience and its overall immersiveness [4,5,10]. We aim to improve this situation by providing both a 3D view of participants and their surroundings, as well as a realistic and intuitive means to look around it.

Figure 1 demonstrates how a user can treat the video conference stream like a window, where moving left and right (and also up and down) provides changing views of the other participant and the remote environment, providing parallax-based depth cues which are linked to the viewer's head position. This provides one of the most powerful monocular depth cues available and results in a simple but convincing illusion of 3D [18]. The effect is more dramatic and convincing when animated and viewed in-person.

978-0-7695-3454-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ISM.2008.12

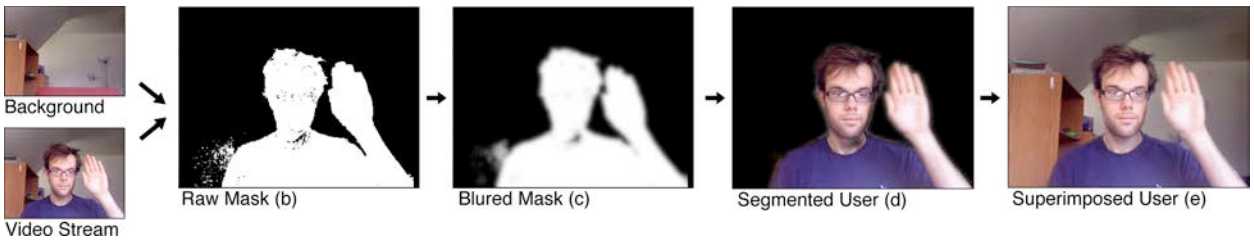

Figure 2. The process employed to extract the video conference participant from the background environment. The mask is blurred to preserve soft edges, like those required for features such as hair, and to reduce noise.

Previous systems have achieved this type of effect with sophisticated sensor and multi-camera setups (e.g., [6,9,15]). The cost, size, and complexity of these systems dramatically limit their potential for widespread adoption [5]. To combat this, our implementation uses only a single generic webcam at each end for both scene capture and the creation of head-coupled pseudo-3D views. In other words, this technique requires no additional hardware beyond what is already required for standard video conferencing: a single webcam. This means the method and its enhanced interactivity and immersiveness are essentially available for free, requiring users to simply download updated software.

2. Implementation

We will first provide an overview of how we capture and create a pseudo-3D video stream and how users interact with the resulting scene. We then discuss the performance of the system, including hardware requirements, accuracy, and tracking speed.

2.1 Creating a Pseudo-3D View

To create a pseudo-3D view using a single conventional webcam, we first extract the participant from the background environment. These separated components are then composited as foreground and background layers with a small offset. This, in concert with a mobile virtual camera coupled to the user's viewing position, creates realistic motion parallax, producing a compelling 3D experience.

Seperation of the user's image is achieved with background subtraction, of which there are numerous sophisticated methods that can be employed (e.g., [3,19,21], see [14] for an overview). A simple per pixel YUV color-space distance comparison was sufficient for our prototype system. This technique requires an unobstructed view of the background (i.e., without the participant) for comparison. There are several ways to capture this, including waiting for when the user is not present, building up a background image over time as the user shifts in their seat, or simply explicitly asking the user to temporarily move out of the way. The latter method is employed by Apple's popular iChat for adding novelty backdrops, such as the Eiffel tower [8] and is used here.

Background subtraction produces a mask of the user (Figure 2b). However, variations in lighting (e.g., shadows) and other camera artifacts inevitably produce at least some noise. To dampen this isolated static and to help fill gaps in the user's mask, we perform a Gaussian blur (Figure 2c). This has the added benefit of softening the edges, which blends the user more smoothly into the background, important for features like hair (where a Boolean mask is too harsh). Figure 2d shows the final result of user segmentation. The halo is highly translucent, and blends into the environment when we reintroduce the background, as seen in Figure 2e. As we will see, the resulting 3D effect allows viewers to look behind the participant.

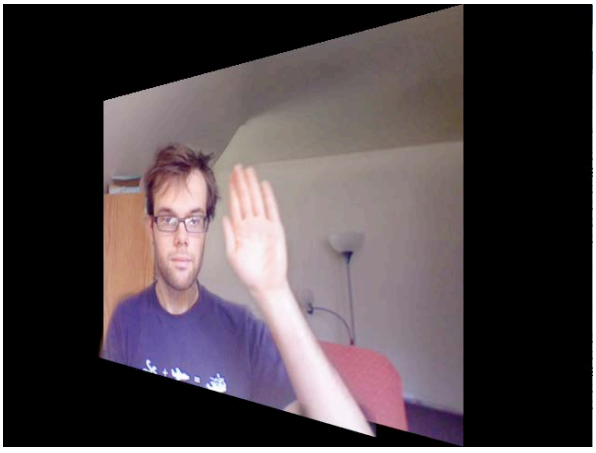

Figure 3. The video steam rendered as two layers and viewed at an extreme angle. The user (foreground) is scaled and offset slightly forward.

This necessitates the use of either all or part of the image captured for background subtraction (as the user occludes background pixels behind them in the video stream).

Once the user is successfully segmented from the background image, we can move the two layers apart, treating the user as the foreground. This offset combined with a head-coupled virtual camera creates appropriate motion parallax – one of the strongest monocular cues for depth [18]. It is also necessary to scale down the user's image to maintain relative size. Figure 3 shows the result of this action at a very oblique angle. The separation of the two layers is most apparent near the elbow.

2.2 Head-Tracking

We employ face tracking to locate a user's head in three dimensions (Figure 4). Our system discards the Z position (roughly equated to lean extent) so as to maintain a fixed video size (early versions allowed the viewer to lean forward to zoom). The X and Y coordinate data are used to position a camera in the virtual environment. As the user moves, so does the virtual camera. So, just as someone would move their head to look through a real window, users can move their head to look through a virtual window on their computer monitor.

Our implementation has two significant benefits. Foremost, because we use face tracking, the user does not have wear any markers or tracking devices. Secondly, the spatial information can be extracted from the same video stream that is sent to the remote participant. This means a single camera can be used to both transmit video to a remote user and track the local user. Thus, no additional sensors or cameras are required.

2.3 Performance and Accuracy

We have shown that we can achieve image capture, background separation and head tracking with the video stream from a single generic webcam, allowing our pseudo-3D video conferencing technique to be deployed entirely in software. However, in order to be feasible, the software must run in real-time on modest hardware, otherwise the 3D benefits are a moot point. To test this, development and experimentation was conducted on an Apple MacBook equipped with a 2.16Ghz Intel Core 2 Duo processor – a current, but modest machine, which will be comparatively underpowered not too long after publication of this paper. This model laptop features a built-in, fixed-focus, 640x480 resolution webcam able to capture video at 30 frames per second.

Figure 4. The video captured by the webcam is not only sent to the remote conference participant, but also used to track the local user's face. The derived positional data is used to control a virtual camera in the pseudo-3D scene.

Our proof-of-concept system is comprised of two concurrent processes. One is a Java-based graphical front-end, seen in Figure 1. JOGL, an OpenGL wrapper for Java, is used for the 3D graphics. Although OpenGL offloads processing to a dedicated GPU whenever possible, our application, essentially two textured planes, can be run entirely in software with acceptable frame rates. We also found that live background subtraction in Java is easily achieved on modest hardware. The biggest performance obstacle was with video capture, which required 30 frames per second of video to be copied into the Java Virtual Machine as bitmaps (for pixel-level processing). We feel that moving to a C/C++ code base would likely alleviate this bottleneck and significantly improve performance. Nonetheless, we achieved real-time performance with our prototype Java implementation on a single processor core (one of two).

Face tracking is the second and final component in our implementation. This proof-of-concept application, written in C++, uses OpenCV's Haar Cascade classifier to identify faces [20]. Although it can be used to detect many faces within a single image, we only track the position of the closest face (i.e., the largest face). This makes it robust against other people looking over the primary viewer's shoulders or simply passing through the background (i.e., head tracking should stay fixed to the primary viewer). The head tracking also proved surprisingly accurate for a wide variety of faces, including those with mustaches, beards, hats, and glasses. In rare cases, intense lighting in the background caused some problems, as the facial features were overwhelmed when the camera auto-adjusted the exposure. This can be overcome, however, by exposure-locking the webcam.

Considerable effort was invested in optimizing our face tracking code, and then finding a suitable balance between performance and accuracy. Our final implementation is able to track faces at 25 frames per second on our test laptop. This consumes less than 50% of the total CPU power (the application, which is multithreaded, is distributed across both processing

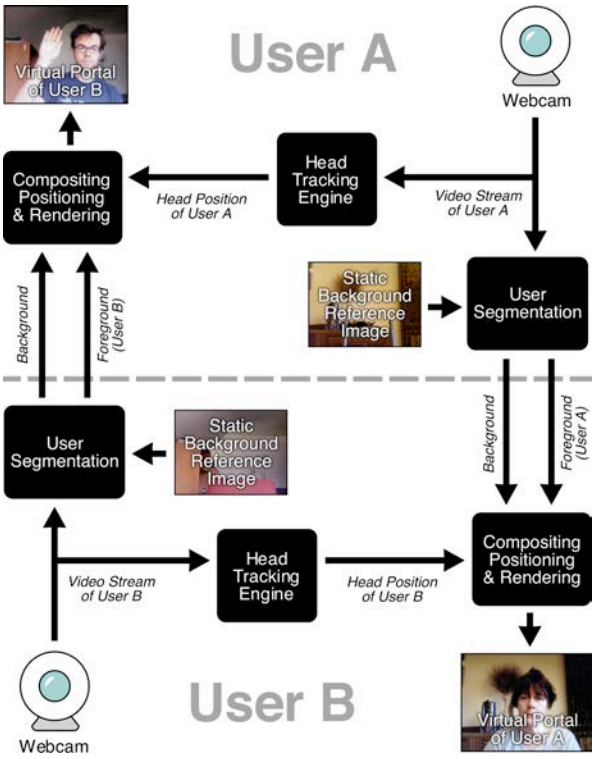

Figure 5. High-level system architecture. User segmentation can occur on the local (shown) or remote computer.

cores). This was acceptable performance for our proof-of-concept implementation, and allowed both our 3D front-end and face tracking components to run in parallel without consuming the entire CPU.

At this level of performance, our face tracking implementation was sufficiently accurate that no averaging or filtering needed to be applied to stabilize the resulting X and Y coordinates. The only requirement imposed was that a minimum movement of two pixels was needed before the virtual camera position was updated. This effectively prevented any pixel-boundary flickering. Because we use the raw positional data from the last captured frame, our latency is around 40ms, which while perceivable, is not distracting and allows for fluid and intuitive exploration of the 3D video conference portal.

Network performance and requirements are similar to that of a traditional video conferencing application. Video can be transmitted to the remote user untouched - only the background from which to subtract the user (for segmentation) is needed. This image could be easily transmitted when the video conference is initiated (and even updated periodically). Alternatively, user segmentation could be performed on the local machine and streamed pre-processed. The head tracking positional data is only used locally, and never has to be transmitted.

Finally, although we did not attempt to port our system to a mobile phone, we believe that the increasing computational power of these devices means they will be strong candidates for this type of pseudo-3D video conferencing in the near future, especially given the lowered resolutions likely mandated by communication bandwidth issues,

3. Related Work

3D video conferencing is not a new concept. Efforts in this domain began decades ago, especially as technologies and computational power emerged that allowed for real-time 3D video capture. Previous systems employed a wide variety of approaches to create a 3D experience.

The simplest option is to not process the live video at all, but rather substitute the user for a 3D avatar, which can be easily placed into a virtual environment (e.g., [1]). However, this approach has obvious shortcomings, most notably the fact that it is not possible to see the remote user. Other systems go a step a further by inserting the video stream into a virtual environment (e.g., [12,16]). In most implementations, this appears as a textured rectangle. Although it is now possible to see the remote user, the environment is decidedly artificial. Also, the heavy contrast between the synthetic environment and the photo-realistic video stream is quite jarring. Movement in this virtual environment is typically achieved with a keyboard and mouse.

Considerably more sophisticated approaches rely on multiple cameras, the data from which can be used extrapolate a 3D representation of the scene (e.g., [2,9,11,22]). Although providing a high degree of realism, the equipment cost and computation requirements are substantial obstacles for broad adoption. Additionally, the large footprint and semi-permanent nature of these systems makes their adoption in the home unlikely. This simultaneously precludes their use in mobile devices, such as cell phones and laptops. Even more problematic is that many of these systems require headgear, for example, head mounted displays or shutter glasses (e.g., [4,6,15]). Not only is this cumbersome, but also highly obtrusive for video conferencing, as the face is obscured.

It is also worth noting that there has been considerable work focusing on head tracking and general 3D interaction; see for example [17].

There are two recent advances that could be used to enhance future versions of our system. Foremost, new techniques are being developed that can create 3D scenes from conventional 2D images by analyzing geometric features (e.g., [7]). Like our present system, this requires only a single, conventional camera. Secondly, there have been great strides in 3D imaging. Light field photography, for example, can capture 3D information from a single exposure [13].

4. Conclusion

In this paper, we presented our implementation of a 3D video conferencing system. This system has the unique property of not requiring any special, additional hardware to create both a pseudo-3D view of the remote user and environment, and to track the head position necessary to provide convincing motion parallax depth cues. Requiring only a single traditional webcam means the technology readily deployable.

5. Acknowledgements

This work was supported in part by grants from the Intel Research Council, General Motors, and the National Science Foundation under Grant IIS-0713509.

6. References

1. Carlsson, C. and Hagsand, O. DIVE - A Multi-User Virtual Reality System. In *Proceedings of the IEEE Virtual Reality Annual International. Symposium, VRAIS '93*, pp. 394-400.

2. Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H. Free-viewpoint video of human actors. In *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '03, pp. 569-577.

3. Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 25, 10 (Oct. 2003), 1337-1342.

4. Fuchs, H., Bishop, G., Arthur, K., McMillan, L., Bajcsy, R., Lee, S., Farid, H., and Kanade, T. Virtual Space Teleconferencing Using a Sea of Cameras. In *Proceedings of the First International Conference on Medical Robotics and Computer Assisted Surgery*, 1994, pp. 161-167.

5. Gaver, W. W., Smets, G., and Overbeeke, K. A Virtual Window on media space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pp. 257-264.

6. Gross, M., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-Meier, E., Svoboda, T., Van Gool, L., Lang, S., Strehlke, K., Moere, A. V., and Staadt, O. blue-c: a spatially immersive display and 3D video portal for telepresence. In *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '03, pp. 819-827.

7. Hoiem, D., Efros, A. A., and Hebert, M. Automatic photo pop-up. In *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '05, pp. 577-584.

8. iChat: http://www.apple.com/macosx/features/ichat.html

9. Kimata, H., Kitahara, M., Kamikura, K., Yashimat, Y., Fujii, T., Tanimoto, M. System design of free viewpoint video communication. In *Proceedings of The Fourth International Conference on Computer and Information Technology*, CIT '04, pp. 52–59.

10. Kishino, F., Miyasato, T., and Terashima, N. Virtual space teleconferencing - Communication with realistic sensations. In *Proceedings of the 4th IEEE International Workshop on Robot and Human Communication*, RO-MAN '95, pp. 205-210.

11. Moezzi, S., Katkere, A., Kuramura, D. Y., and Jain, R. Immersive Video. In *Proceedings of the IEEE Virtual Reality Annual international Symposium*, VRAIS '96, pp. 17-24.

12. Nakanishi, H., Yoshida, C., Nishimura, T., and Ishida, T. FreeWalk: a 3D virtual space for casual meetings. *IEEE Multimedia*. 6, 2 (Apr-Jun 1999), 20-28.

13. Ng, R., Levoy, M., Br, M., Duval, G., Horowitz, M., Hanrahan, P. Light Field Photography with a Hand-held Plenoptic Camera. *Stanford University Computer Science Tech Report*, CSTR 2005-02, April, 2005.

14. Piccardi, M. Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, SMC '04, pp. 3099-3104.

15. Prince, S., Cheok, A.D., Farbiz, F., Williamson, T., Johnson, N., Billinghurst, M., and Kato, H. 3D live: real time captured content for mixed reality. *In Proceedings of the International Symposium on Mixed and Augmented Reality*, ISMAR '02, pp. 7-14.

16. Regenbrecht, H., Ott, C., Wagner, M., Lum, T., Kohler, P., Wilke, W., and Mueller, E. An augmented virtuality approach to 3D videoconferencing. In *Proceedings of The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pp. 290-291.

17. Rekimoto, J. A vision-based head tracker for fish tank virtual reality-VR without head gear. In *Proceedings of the IEEE Virtual Reality Annual international Symposium*, VRAIS '95, pp 94-100.

18. Rogers, B. and Graham, M. Motion Parallax and the perception of Three-Dimensional Surfaces. In *Proceedings of the NATO Advanced Study Institute on Brain Mechanisms and Spatial Vision*, 1983, pp. 95-111.

19. Stauffer, C., and Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In *Proceedings of Computer Vision and Pattern Recognition*, CPVR '99, pp. 246-252.

20. Wilson, P. I. and Fernandez, J. Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges*. 21, 4 (Apr. 2006), 127-133.

21. Wren C., Azarbayejani, A., Darrell, T., and Pentland, A. Pfinder: Real-time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19, 7 (1997), 780-785.

22. Würmlin, S., Lamboray, E., Staadt, O. G., and Gross M. H. 3D Video Recorder: a System for Recording and Playing Free-Viewpoint Video. *Computer Graphics Forum*. 22, 2 (June 2003), 181-193.