# The Vis-a-Vid Transparent Video Facetop

*David Stotts, Jason McC. Smith, and Dennis Jen*
Dept. of Computer Science
Univ. of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175 USA
{stotts,smithja}@cs.unc.edu

## OVERVIEW

The Vis-a-vid (VAV) transparent video facetop is a novel video-based user interface that supports not only single-user interactions with a PC, but also close pair collaborations, such as that found in distributed pair programming and distributed extreme programming (dPP/dXP) environments. In VAV we form the facetop by pointing an inexpensive but high quality iBot camera at the PC user, and then making that live video stream fullscreen in size. In software, and using a high-performance 3D-graphics video card, we reverse the video image horizontally, make the video window semi-transparent and send it to the back of the video-rendering list for applications on the desktop. This gives us a desktop "live video wallpaper" with the appearance of the user watching his/her own desktop from behind.

**Figure 1: Vis-à-vid video facetop (iBot in foreground)**

Using image analysis techniques, we track the users fingertip in the backing window, and drive the mouse from this analysis application. The user can then manipulate the PC desktop as he would with the mouse. The VAV concept is most effective with wall-size PC desktop projection, but works well with traditional PC monitors also. In a projected environment, the user experiences the effect of manipulating the desktop from his seat by pointing at the projected image. Seeing the self-image gives unique visual cues for pointer control not found in other user interfaces. We exploit these visual cues for close synchronous collaborations by compositing two facetops on a shared desktop. Both users see each other, the same desktop, and where each is pointing.

This work solves the "pointing problem" common to collaborative software applications. Using the facetop, users can see each other as they collaborate, and concurrently see their shared work and make visible gestures relative to that work. The facetop provides a novel alternative to the classic WIMP desktop for single-user PCs. Figure 1 shows this application for Windows. Our demo is a second prototype, a Mac application.

## RELATED WORK

Space does not allow a full overview of related work, but a few projects should be mentioned. Giving your PC "eyes" is a growing concept, as is illustrated by this 2001 seminar at MIT [4]. The VAV exploits this notion. The VAV PC has eyes (the iBot camera) and watches the user's actions. The user sees him/her self in the desktop, however, and this visual feedback gives the cues needed for effective PC control. The user's visual presence is settable dynamically, from very faint (transparent) to very opaque. Different levels of transparency have different natural uses at different times in PC control and collaboration.

Many researchers have made systems that have transparent tools, windows, pop-ups, sliders, widgets that allow see-thru access to information below; these are primarily used for program interface components [1,2]. Many systems have some user embodiment and representation in them (avatars), especially in distributed virtual environments like [3], but these tend to be generated graphics and not live video. Our work also uses gesture recognition results, e.g. [5,6]. The missing concept in current gesture work is the immersion of the user into the PC environment. In other work the user is still an object separate and apart from the PC being interacted with. In VAV, the user is given the illusion of being part of the environment being manipulated.

In the domain of collaborative systems, C*learBoard* [7] is especially applicable to our research. It was a non-co-located collaboration support system that allowed two users to appear to sit face to face, and see the shared work between them. The ClearBoard experiments showed that face-to-face visibility was enhancing to collaboration effectiveness, but the workstations were expensive and custom built. The facetop uses ubiquitous off-the-shelf PC equipment and a $100 iBot camera.

## SYSTEM IMPLEMENTATION

Refer to Figure 1 for the prototype VAV screen This was generated with a WinPC and high-end graphics card to perform the image transparency. Performance was slow, and for the second working facetop prototype, we designed for the Apple Mac platform (which has better integration and access to the OpenGL layer). VAV is implemented on MacOS X 10.2 by taking advantage of the standard Quartz Extreme rendering and composition engine. QE renders every window as a traditional 2D bitmap, but then converts these to OpenGL textures. By handing these textures to a standard 3D graphics card, it allows the highly optimized hardware in the 3D pipeline to handle the compositing of the images with varying transparency, resulting in extremely high frame rates for any type of image data, including video blended with the user interface.

The video signal is generated from a Sony iBot camera, at 640 x 480 pixels in color with no compression, at 30 frames per second. The iBot was set up 2 feet from the user at the keyboard, pointing back at the user. Note that the VAV facetop will work in principle from stored video as well as live. Tracking is done on the video frames no matter how they are generated, and signals will go to the mouse driver irrespective of video source.

The video application, with tracking capabilities, is run in a standard MacOS window, set to full screen size. Using OpenGL, setting the alpha channel level to near-transparency gives the faint user image we need. Placing the application to the back of the rendering list allows the desktop icons to be on top in full image strength. OpenGL rendering in MacOS does compositing of the VAV image with the desktop.

### *Single User Applications for VAV*

The primary VAV use is the source of the term "facetop". We use a transparent video window, full screen, and a backer for the entire desktop. A user sees him self as a "ghostly" image apparently behind the desktop, looking back at the icons and windows from behind. Instead of a desktop, we see a "*face" top*. The Ovaltine tracker library we developed for an earlier application (hyperlinks in real-time video) is used to look for and track the fingertip. In noisy environments (busy backgrounds) we boost performance to acceptable frame rates by having the user wear a special thimble on the finger; allowing a chromatic transformation of each video frame into an image where the tip is simple and fast to find. In addition to normal desktop/mouse manipulation, the facetop is useful for web browsing with browsers like Opera that allow "mouse" gestures such as wiping left to cause a backup function.

### *Collaborative Use of the VAV*

An equally interesting domain of application for VAV is in collaborative systems, specifically, synchronous pairing applications. We have been investigating such a system for the past year for use in distributed Pair Programming and distributed Extreme Programming (dPP/dXP). Pair programming is a software engineering technique where 2 programmers sit at one PC to develop code. One types ("drives") while the other reviews and assists ("navigates"); roles swap frequently. We do this with a non-colocated user pair via desktop-sharing software like pcAnywhere and NetMeeting.

For dual-user VAV, the facetop gets shared along with the entire desktop. Both video streams (each collaborator) get composited and superimposed on the shared desktop. Each sits so that the two heads are on different sides of the frame. The video image is flipped so that when one collaborators points at a desktop icon, the other partner sees him pointing to the same icon (of course). In this "knitted together" joint image, we sit each user on a neutral background to make the VAV image unobtrusive.

We expect the VAV facetop to give them a new, additional capability, which is to point without having to pass control of the mouse, so the other person can keep typing. We are experimenting to see if adding this capability makes programming easier, more productive, or more pleasant of an experience.

## CURRENT INVESTIGATIONS

The natural extension to the basic VAV concept of finger tracking is gesture recognition. We are expanding the tracking capabilities to allow a broader range of user actions to be recognized and tracked. Some simple gesture recognition will be needed in the first prototypes. Once a user has pointed at an icon on the desktop, and the VAV has tracked the fingertip to the location, the VAV must also recognize some movement, action, or gesture as indicating actions other than mouse clicks

### *Questions under investigation*

- How effective is the VAV facetop as a mouse-replacement in a traditional WIMP desktop?
- What is the most effective camera angle and placement for comfortable arm movement and hand motion?
- How does VAV effectiveness compare in a projected environment vs. a CRT-based environment?
- Do distributed pairs perform programming tasks better with VAV facetop?
- Are there uses for the VAV in multi-user environments other than paired collaborations?
- Can we use a broad range of user gestures in the VAV for desktop and program control?
- How can the VAV be technically implemented for different platforms with different levels of graphics support?

**Acknowledgements** This work was partially supported by a grant from the U.S. Environmental Protection Agency, # R82-795901-3.

## REFERENCES

[1] Eric A. Bier, Ken Fishkin, Ken Pier, Maureen C. Stone*, "*A Taxonomy of See-Through Tools: The Video, Xerox PARC, Proc. of CHI '95, http://www.acm.org/sigchi/chi95/Electronic/documnts/videos/eab1bdy.htm

[2] Beverly L. Harrison , Hiroshi Ishii, Kim J. Vicente, and William A. S. Buxton,"Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention," Proc. of CHI '95, http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/blh_bdy.htm

[3] Steve Benford, John Bowers, Lennart E. Fahlén, Chris Greenhalgh and Dave Snowdon, "User Embodiment in Collaborative Virtual Environments,", Proc. of CHI '95, http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/sdb_bdy.htm

[4] Vision Interface Seminar, Fall 2001, MIT, http://www.ai.mit.edu/~trevor/6.892/

[5] T. Nishi, Y. Sato, H. Koike, "SnapLink: Interactive Object Registration and Recognition for Augmented Desk Interface," Proc. of IFIP Conf. on HCI (Interact 2001), pp. 240-246, July 2001.

[6] Bretzner, L., and T. Lindberg, "Use Your Hand as a 3-D Mouse," Proc. of the 5$^{th}$ European Conf. on Computer Vision, LNCS *vol. 1406*, pp. 141--157, Springer Verlag, June 1998.

[7] H. Ishii, M. Kobayashi, and J. Grudin, "Integration of inter-personal space and shared workspace: ClearBoard design and experiments," *Proc. of ACM Conf. on Computer Supported Cooperative Work*, Toronto, 1992, pp. 33-42.