Integrated video communication with videoSpace

Nicolas Roussel
Laboratoire de Recherche en Informatique
UMR 8623 CNRS - Université Paris-Sud
91 405 Orsay Cedex, France
roussel@lri.fr

Media spaces were designed to support informal as well as formal communication among remote users. We believe that these systems must be integrated into existing environments in a lightweight way, blending support for collaboration with the user's existing computer environment. This demonstration presents *videoSpace*, a software toolkit that facilitates the integration of live video into documents and applications.

Introduction

VideoSpace is a software toolkit designed to facilitate the integration of live video into new or existing documents and applications. Inspired by previous work on Media Spaces (Bly et al., 1993), VideoSpace promotes the development of collaborative environments where communication facilities are embedded into the documents and applications rather than provided as separate applications. VideoSpace also supports novel uses of video, e.g. video as input device, by supporting real-time filtering of live video images. Finally, since the video streams managed by videoSpace often involve live video of people, videoSpace takes privacy issues into account.

Embedding communication facilities in documents and applications

The first user-level component of videoSpace is *videoServer*, a personal server run by the user of a workstation. It uses the HTTP protocol to make still images and video accessible to other users through simple URLs. Video data is sent to the client on a separate UDP connection, or on the HTTP connection itself, as a server-pushed series of JPEG-compressed images. This allows some Web browsers (most notably Netscape) to display video in any window, without any plug-ins. Users can thus create their own media space and customize their interface by copy-pasting pieces of HTML to embed live video into any Web document.

The second user-level component is *videoClient*. VideoClient takes advantage of the architecture of the X Window system to display video in top-level windows as well as subwindows of existing applications. This has been used to "augment" existing Unix and Tcl/Tk applications, to create interfaces to other videoSpace components and to integrate them with GroupKit (Roseman & Greenberg, 1996).

Supporting novel uses of video

VideoSpace is implemented in C++. Developers can use its API to create more complex applications, including multiparty video communication services and applications that use external analog monitors. VideoSpace also supports video filters that transform and analyze video frames in real-time. Image transformation filters include cropping, resizing and gamma-correcting images. Image analysis filters include image-differencing and chroma-keying. We have used image-differencing to detect motion and chroma-keying to overlay live video onto running applications. This has proved useful to support lightweight interaction.

Supporting privacy

When videoServer receives a connection from a client, it runs a notification and control script that controls access by that client, adjusts the type or parameters of the request and notifies the user. Access control is typically based on the client's machine name and other available information, such as the user's login name. Modification of the request can be used to send a pre-recorded video clip or a lower-quality image to unknown clients. The notification can be visual or auditory and may have arbitrary side effects. For example, it can create a bi-directional video link by opening a second connection back to the caller.

Conclusion

VideoSpace is in daily use in several laboratories around the world (see http://www-ihm.lri.fr/~roussel/videoSpace for availability) and its flexibility has already proven useful. Future work includes extending the toolkit, learning from both programmers and end users and experimenting with new real-time video filters.

References

Bly, S., Harrison, S. and Irwin, S. (1993) 'Mediaspaces: Bringing people together in a video, audio and computing environment'. *Comm. of the ACM*, 36(1):28-47.

Roseman, M. and Greenberg; S. (1996). 'Building Real Time Groupware with GroupKit, A Groupware Toolkit'. *ACM Trans. on Computer-Human Interaction*, 3(1):66-106.