

GV-STC Video Widgets

October 1993

Abstract

This document describes the design and function of a suite of software tools developed by the GV-STC in support of its digital media infrastructure. These tools provide close integration of graphics workstation platforms with video production and conferencing environments. Standard hardware provides for direct serial control of video signal switching, video tape transports, robotic cameras, and conferencing equipment. The software system provides command-line and graphical user interfaces to the video devices, and it incorporates an extensible scripting system for coordinated control of multiple devices. The software is UNIX-based, following a client-server model, and it uses Motif as the graphical interface toolkit. This standards-compliant implementation allows remote control any of our video devices from graphics workstations anywhere on the Internet. It also allows groups with similar commercially-available hardware to immediately make use of the software tools and applications.

1 Introduction

We at the GV-STC have developed a suite of software tools to allow closer cooperation between graphics workstations and video equipment than previously possible. The impetus for developing these tools was the demand for coordination and control of diverse video and computer-based resources that was prompted by the GV-STC's acquisition of a full-time, dedicated T1 network for video and data transmission that is used to support Center-wide teaching, research collaboration, and administration.

The software tools described in this document are part of a larger project undertaken by the GV-STC to develop a digital media infrastructure to support its activities. This infrastructure is designed both to satisfy the Center's immediate communications needs and to allow incorporation of new and emerging computing and communications technologies. References in the attached bibliography document related projects in this larger effort.

2 Hardware Overview

Figure 1 illustrates the hardware environment in which the video tools operate. Fundamentally there are two worlds: the analog world of video production and the digital world of computer graphics workstations. Bridging the worlds are several kinds of devices. On the workstation side are video input cards that digitize analog video signals, which allows the resulting image files to be stored and manipulated digitally. On the analog video side are scan converters which resample the high-resolution video output of computer workstations to produce signals that can be recorded on videotape. There are also hybrid devices like the digital disk recorder, which looks like a file system to a computer workstation (it accepts files of image data via network protocols), but like a video tape recorder to a video device (it can use the images as frames in an analog video signal).

Apart from the fundamental analog/digital split, one problem that has prevented the closer integration of the video and computer graphics worlds is a fundamental difference in the approaches that the video and computer communities have taken to device communication and control. In the video community device control protocols are typically proprietary and specific to one manufacturer's product line. This allows a manufacturer to produce tightly integrated systems, but it hobbles the development of heterogeneous environments. In contrast, in the computer industry, standard communications media and protocols have fostered an environment where nearly-transparent access to networked resources is a reality.

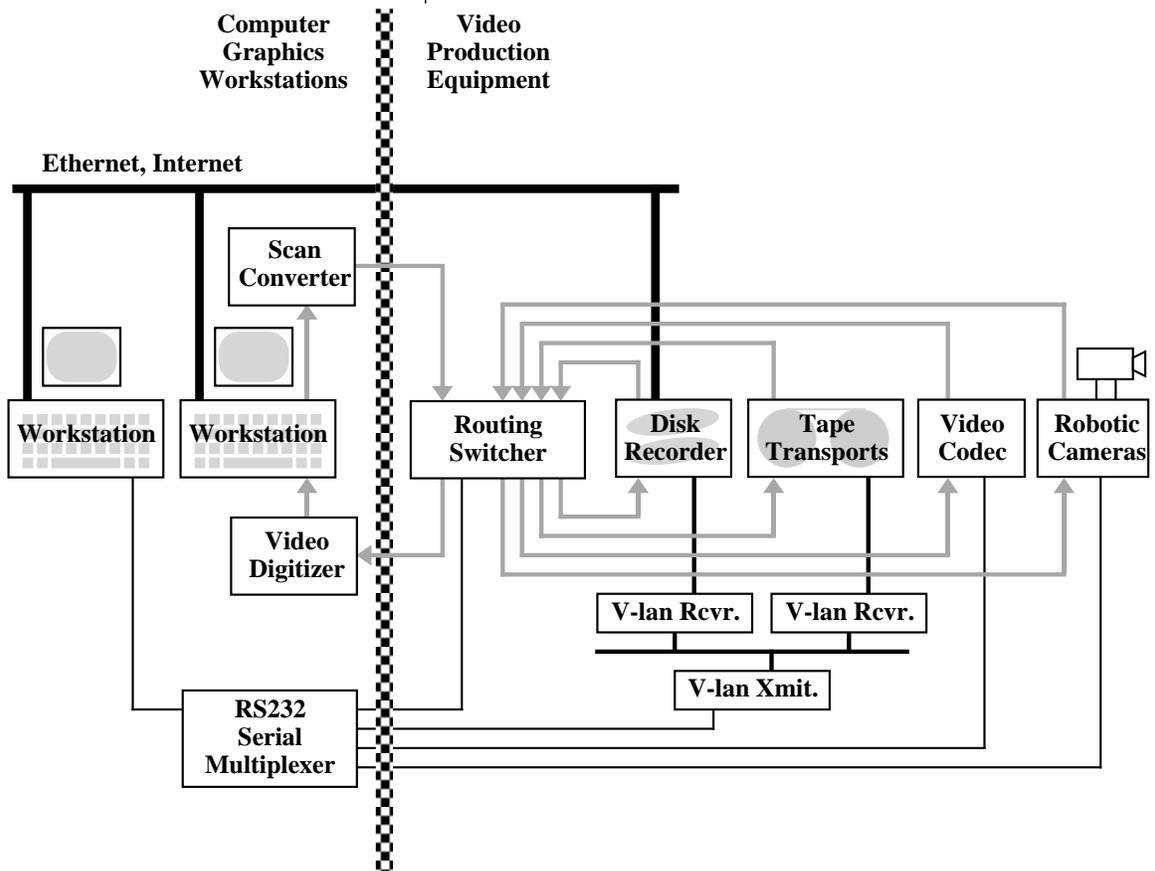


Figure 1: The separate worlds of workstations and video.

Since our goal in this project was to provide network-based control of video resources, a fundamental requirement for the hardware design was that all devices be able to communicate with workstations using RS-232 serial protocols. This allowed us to write modular device drivers which shielded the applications interface from hardware dependencies and fosters extensibility and portability.

3 Device Control

There are four classes of video devices that the video tools were designed to control:

3.1 Routing switchers

A routing switcher is an electronically controlled patch bay. Multiple video and audio inputs can be simultaneously connected and switched to multiple outputs. Our routing switcher has 16 inputs and outputs on four “levels”: a “component” level for high-resolution RGB workstation video and Betacam signals; a “composite” level for standard NTSC-encoded video; and two levels for stereo audio signals. The video tools can easily be configured to accommodate routers with different features.

3.2 Transports

Transports are either video tape recorders (VTRs) or disk or solid-state devices that emulate VTRs. Since VTR manufacturers still use proprietary protocols to address their transports, we introduced an intermediate layer of device control hardware between the workstation and the transport based on the rapidly developing V-LAN standard. This gave us a standard control protocol to address all our tape and disk transports. V-LAN’s transmitter-receiver model allows the control system to be reconfigured for different transports by simply downloading a different driver to a transport’s receiver.

3.3 Robotic cameras

Remotely controllable camera units are a standard feature of videoconferencing and distance learning environments. As with transports, systems to control the pan, tilt, zoom, and focus of the cameras are largely proprietary and designed to be used in conjunction with complex room control systems that typically don’t offer direct computer control of functions. We based our camera control functions on a new system introduced by ParkerVision of Jacksonville, Florida. The camera units offer infrared remote and standard RS-232 serial control of all camera functions, and they also have a unique feature that causes the camera to automatically track a moving speaker wearing a wireless microphone.

3.4 Multimedia conferencing systems

The GV-STC has been actively using its T1 network for research collaboration and teaching. Meetings and presentations are frequently multimedia events that include live and graphics camera inputs, slide transparencies, presentation graphics systems, video tapes, interactive demonstrations on graphics workstations, projection systems, and video inputs from remote sites within the GV-STC. In order to coordinate the media used in these productions, a single operator needs to be able to control the whole system, and we need the ability to automate common tasks. To provide this kind of control, the video tools incorporate a scripting system that allows simultaneous and coordinated control of router, transport, and camera functions. The tools provide both a manual, button box interface and a similar graphical interface to complex multimedia control functions.

4 Software Applications

The GV-STC has written a set of independent but cooperating programs to take advantage of the remote control features outlined above. They run portably under the standard UNIX and Motif environment. All these programs work from any workstation on our network, or, transparently to the user, from elsewhere on the Internet.

4.1 Router Controller

The router control program, called `ROUTER`, lets the user make connections between the various pieces of video equipment in the system. It uses a command-line user interface to allow both automated and interactive control. A simple command is, `beta1 to program`, which would cause the output of Betacam VTR #1 to appear on the program monitor. In more complicated situations, `ROUTER` will choose the appropriate connection from a set of possibilities: routing a device to `projector` results in connections to different channels of our multi-format video projector depending on what type of signal the device generates. Because the interface uses symbolic names for devices, the physical connections can be moved without any need to notify users or update programs that call `ROUTER`; they need only use the name `beta1` to refer to the Betacam deck, no matter which input or output of the router it's connected to. `ROUTER` has status commands to inquire about connections that exist, as well as to set them. It can also take a snapshot of the router's entire state and save it to a file, which can later be used to restore the same configuration.

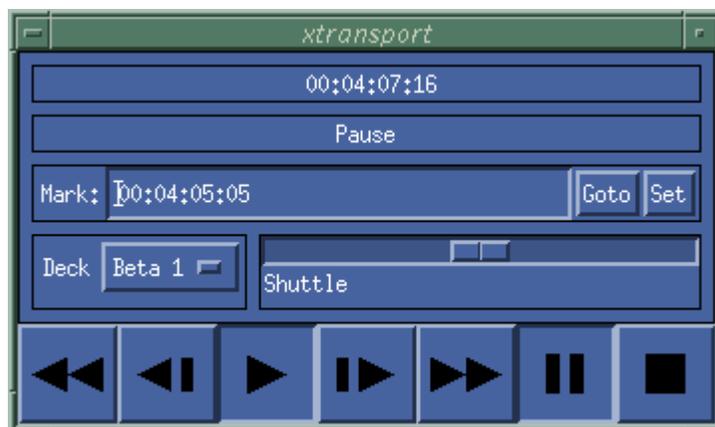


Figure 2: The XTRANSPORT program.

4.2 Transport Controller

Our video software also includes a transport control program, called XTRANSPORT, which gives a uniform graphical interface to all the transport devices in the system. The user interface, shown in Figure 2, is designed to work similarly to the controls of a typical tape transport or VCR: buttons start the tape playing, pause it, stop it, jog forward and backward one frame at a time, etc.; the *shuttle* slider provides variable speed control to search forward and backward for a particular event on the tape. Like a professional video deck, XTRANSPORT gives a continuous indication of the tape's position (in hours, minutes, seconds, and frames) and the transport's status (playing, stopped, turned off, etc.); we go beyond typical devices in that the user can type a time into the *mark* text widget and search for it (this feature can also be used to remember a position and return to it). The *deck* menu changes which transport is being controlled (the controls always work consistently, independent of the transport XTRANSPORT is controlling).

4.3 Camera Controller

To take advantage of the camera control hardware, we have written a program called XCAMERA, shown in Figure 3, which provides an intuitive user interface to the robotic camera. The two-dimensional “joystick” widget at the left controls panning and tilting, and the sliders at the right give control over lens functions. XCAMERA is especially useful in conjunction with a live video program that shows the output of the camera on the same display as XCAMERA's window. Like the other programs, XCAMERA can be run across the Internet, giving one site an active “window” into another site's laboratory.



Figure 3: The XCAMERA program.

4.4 Centralized Control for Conferences

During a conference or live video presentation, a user typically doesn't need all the functions provided by the GUIs, but rather needs fast, simple access to functions commonly used during conferences. In addition, since presentation transitions often involve multiple destinations (when the presenter wants to show a video tape, the image should be projected for local viewers, sent across the video network to remote viewers, and sent to the VTR that is recording the event), it is desirable to be able to perform sets of actions simultaneously. To this end, we use a button box, connected to a workstation, to make simple functions easily accessible. A program runs at all times, taking input from the button box and, when a button is pressed, performing some action associated with that button. These actions can perform any of the functions of the controlled devices, and can even run UNIX programs or scripts. We also wrote a companion program that provides all the functions of the button box from a window (Figure 6). The button box in our laboratory is pictured in Figure 4, and the buttons' labels are repeated more clearly in Figure 5. The top row of buttons switches camera sources between two live cameras, a graphics camera, and a slide projector, both for remote viewers and, in the last two cases, for local viewers watching the projector in the conference room. The first four buttons in the second row select transports for both local and remote viewers, and the four buttons below them control the most recently selected transport. The four buttons labeled *HP*, *Mac*, *DEC*, and *NeXT* select workstation displays to be scan-converted to video resolution and shown locally and remotely. The fourth and fifth rows are devoted to controlling cameras, and the bottom row is designed to interface to the teleconferencing functions provided by the GV-STC's videoconferencing control software described in a companion document.



Figure 4: The Button Box. This device provides a direct physical interface to many of the same functions provided by the GUIs. The buttons can also execute macros, so complex presentation functions can be controlled by pressing a single button.

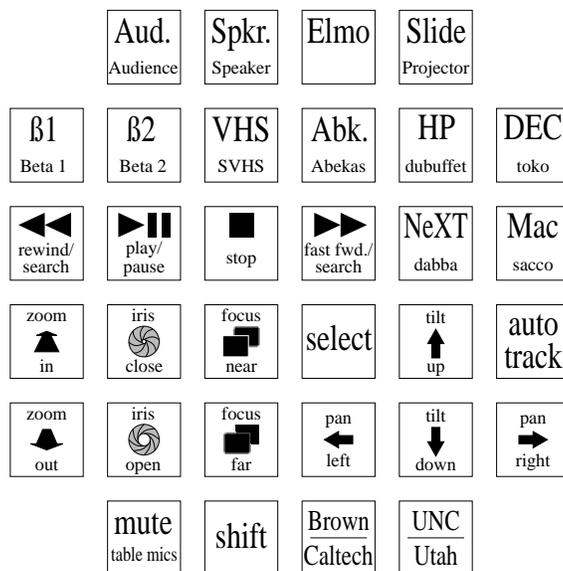


Figure 5: The functions of the Button Box. The four buttons in the bottom row interface to the GV-STC's conference control software (VCC).

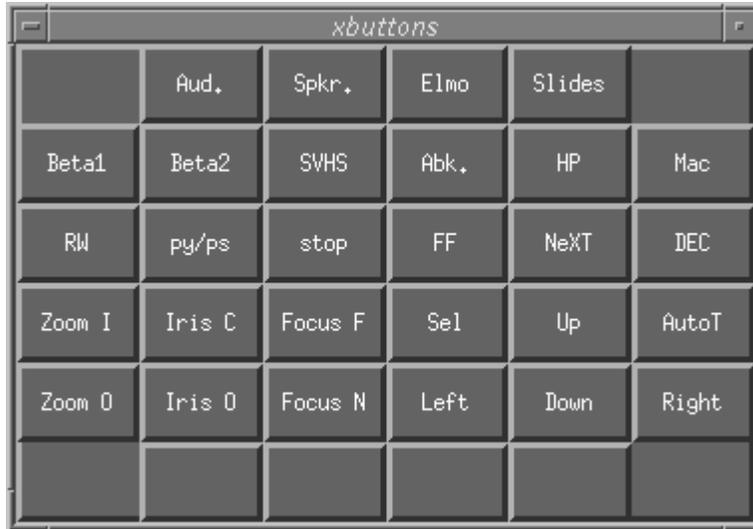


Figure 6: The X version of the button box.

4.5 Edit Controller

Another transport control application, called XEDIT, combines two XTRANSPORT widgets, with extra fields to set edit points, into a single interface that controls the editing capabilities of the V-LAN system. It lets workstation users accustomed to graphical interfaces assemble simple video programs without learning the arcane systems that are the rule in the video world. As shown in Figure 7, XEDIT lets the user set the starting (“in”) and ending (“out”) points on the source and destination transport; the *insert* menu chooses which tracks (video or audio) will be copied during the edit. Since the V-LAN system has frame-accurate control over editing, video programs made with XEDIT are as seamless as ones produced with hardware edit controllers.

5 Software System Architecture

Because the communication lines to our video devices are connected to a single workstation, but we want transparent access to the resources from anywhere on the network, some scheme for network communication is needed. We chose a simple client-server model, using the Remote Procedure Call protocol to simplify communications. Several servers, one for each type of device to be controlled, run on the workstation that communicates directly with the devices; each server accepts simple requests from clients to pass commands to the devices it controls. Since there is at most one server for each serial connection, there is no danger

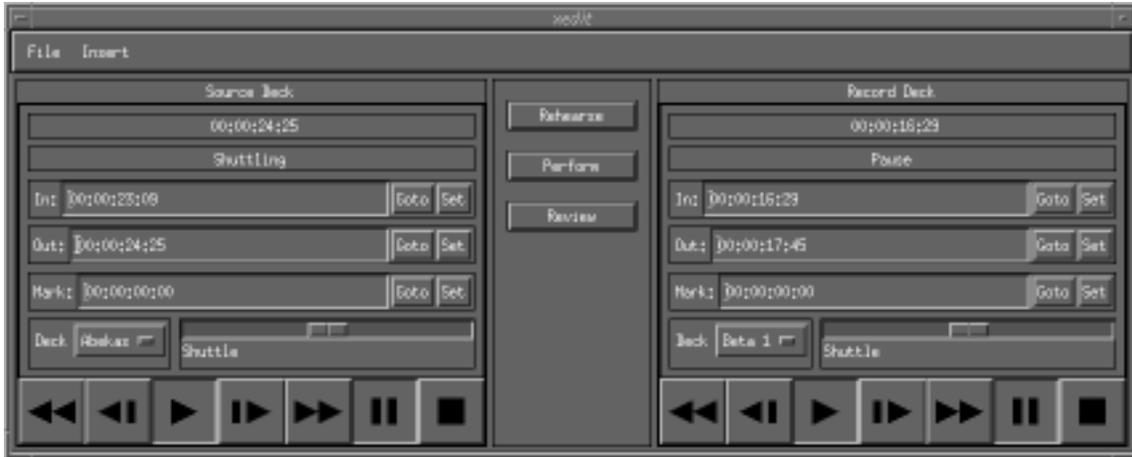


Figure 7: The XEDIT program.

of communications from different clients interfering with one another. Each command that is sent is executed as a unit: the next command isn't processed until a response is received from the device to indicate that the previous command has completed. Because there are several servers, we take advantage of the workstation's multitasking operating system: commands for one device don't have to wait for unrelated commands executing on other devices. Another advantage of having separate servers is that the various devices are independent, so failure in communicating with one device doesn't affect communication with the others.

6 Conclusions

We have described a workstation-based software system, developed by the GV-STC in support of its digital media infrastructure, which controls video devices across computer networks with the flexibility and standard user interfaces to which users of graphics workstations have become accustomed. The traditionally separate video and computer graphics worlds are rapidly converging, but total transparency and interoperability of video and computer-generated media is still some time away. The video tools developed in this project address the need for a greater level of integration between current video and computer technologies, and provide an extensible base on which to build future tools and systems.

7 Acknowledgements

This work was supported in part by the NSF and ARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219). All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.