Video Production in a Visualisation Environment

W.T. Hewitt   Martin Preston

Computer Graphics Unit,
Manchester Computing Centre,
University of Manchester.

Abstract

Computer Graphics provides an extremely powerful tool for information visualisation, and has transformed many subject areas. However, whatever the developments in graphics hardware, it is always possible to produce a program which cannot operate at adequate speed. By using video as an intermediate medium it is possible to visualise the results of such programs efficiently and conveniently. This paper presents the design and implementation of a video production environment, which allows video equipment to be integrated with more conventional graphics systems, as a tool for the computer graphics user.

1 Introduction

The increased availability of high powered computer graphics workstations has transformed the ease by which complicated phenomena can be visualised. However, as the computational power has increased, so has the complexity of information researchers wish to work with. Many applications require the ability to realise large quantities of temporally critical information, which need a graphics bandwidth beyond the most advanced systems.

The only practical solution to this problem is the production of video or films which allow a user to examine the visualisation in real time, without requiring the individual images to be produced at that rate. Unfortunately professional video production is an expensive and complicated enterprise, and beyond the reach of most academic users.

In an attempt to increase the ease by which UK academics can produce videos the JISC NTSC, AGOCG & the University of Manchester have provided funds to establish a "*National Videotape facility*", based at the University of Manchester Computer Graphics Unit (CGU). This facility is intended to allow users to create videos remotely, through JANET, or locally, by visiting Manchester.

In order to assist users in the production of video footage the Manchester CGU has developed a suite of software which handles much of the low level implementation specifics, providing a user with a consistent pleasant interface. This paper is intended to introduce the reader to this environment, hopefully providing him or her with enough information to begin creating exciting videos.

Whilst this paper concentrates on the Manchester implementation, much of the material is also relevant to other environments, which, thanks to the widespread adoption of multimedia, are likely to become available to most graphics producers in the near future.

Following a summary of the fundamentals of European video formats in Section 2 we discuss the video hardware based at Manchester in Section 3,and its comparative merits. Video technology is a complicated area, and for this reason we concentrate only on those aspects likely to most influence the quality of footage which a user may create.

Before we can discuss the high level interfaces provided to users we first outline the functionality of the low level video control software developed (Section 4.2), with particular reference to its portability to other video hardware, and the features it provides. This software is an implementation of a logical image processing pipeline, which allows a user to tailor his or her animation to the peculiarities of the European video standards, as outlined in the previous sections. Particular attention is also paid to the ability to create digital video files (MPEGs), which may be a convenient alternative to true video.

Whilst the low level software is of interest, a user is more likely to be concerned with the interface he or she sees to the process. Section 5, then, presents the developed application environments.

Finally we review the video production process in section 6

2 Fundamentals of Video

Both the creator and audience of any computer generated video will be accustomed to watching television. Viewers expect to see pictures which appear to be moving photographs, packed with detail and composed of vibrant colours. HOwever, in reality the television signal received by the domestic TV is, when compared with the digital images produced by contemporary computers, of extremely low quality, with a poor range of colours and extraordinarily limited resolution. Why, then, are we so happy with TV pictures?

Essentially the production of acceptable quality television has been the product of a long line of ingenuous engineers and researchers, who have, over the years, managed to cram an awful lot of information into a very low bandwidth signal. This has been achieved by concentrating effort into characteristics of pictures which are of most interest to a human viewer, and throwing away large quantities of data. For example, computer graphics users are accustomed to working with “true colour” 24 bit representations of images. Each element of the television picture though, can be comfortably encoded into 8 bits, with 10 now being the norm (to allow for some error correction).

The result of this is that a naïve computer graphics user will produce images which appear fine on a computer screen, but which look dreadful when sent to a TV. The situation is worsened still further by a wide range of television recording media, many of which reduce still further the quality of images displayed.

Does this mean that video is unsuitable for computer graphics? The answer to this is a qualified no, as provided the images we wish to record are produced with the limitations of video in mind, we can generate acceptable recordings.

The topic of this section, then, is the elements of television and video technology which impact on the quality perceived by the viewer. We will concentrate only on those factors which the user can counteract, and hopefully provide the reader with enough information to enable him or her to account for the ravages of the recording process!

2.1 The History of Television

Television has been around for a long time, and throughout its history the overriding research and development maxim has been backwards compatibility. This means that in order to understand the current restrictions of the medium we must outline the development history of TV.

The first television signals were, of course, grayscale images. These analogue signals were encoded at a television studio, then broadcast to low cost decoders (the TV units) which were capable of displaying images relatively quickly. This signal was designed to allow the cheapest possible technology to view these pictures, and this has had an impact on its design ever since. For example the signal is not a continuous stream of intensities, as it had to allow time for the technology to reset itself to begin scanning the screen again.

Of particular importance to the CG user is the interesting decision by the standards authority not to require the edges of the screen to be visible by the viewer. Consequently manufacturers were able to use low quality tubes, which (a) tended not to display the full picture, and, (b) were incapable of producing sharp images at each side of the image. However by placing covers over the screen such aberrations could be masked! This practice, called the 'safe area', continues to this day, so it is industry practise to not place any important information on the surrounding 10% of the image. Obviously this severely effects the freedom of the CG user to position objects on the screen!

The speed at which the images are replaced on a TV screen directly affects the perceived quality of motion. If the refresh is too slow, fast moving objects may appear to strobe. The quality of technology available in the early years directly effected the resolution at which images could be produced, so in order to sustain the required refresh rates it was necessary to limit the quality of pictures. Various competing resolutions are still in use, though the trade-off between quality and resolution is still the overriding concern. The PAL screen which UK users will be concerned with has, assuming a full bandwidth signal is being broadcast, a resolution of 768x575 pixels with a refresh rate of 25 frames per second.

Although 25 frames a second are transmitted, this refresh rate is just on the threshold of motion perception. To achieve moving pictures, as opposed to strobing pictures, a system called interlacing was developed. Each frame of the video signal is composed of 2 fields. Each field contains half of the horizontal lines in the picture, which are shown sequentially. This can cause problems for CG material, which is typically viewed on a non-interlaced computer display. For example, if a line appears only on one video line in one field, it will flicker. Obviously this makes recording of conventional wire frame graphics problematic, and so wider lines are normally used.

The black & white television signal is encoded with a carrier signal, and changes in the intensity of light visible at any point on the screen are represented by changes in the signal. These changes are, of course, continuous signals, unlike the digital data we are more familiar with. This means that the TV signal finds it impossible to represent all the possible changes in images which we may wish to transmit. For example the television cannot represent a black and a white pixel adjacent to one another, as the signal cannot leap from one extreme to the other instantaneously.

This restricts the range of images we may encode, and may cause unpleasant artefacts to appear in videos, such as fringes to appear around sharp edges.

When the engineers began to develop colour signals it was necessary to ensure backwards compatibility with the older B&W receiving sets. This meant that the colour signal must still be viewable with older sets without perceived loss of quality. In order to achieve this the admirable goal of true colour television was sacrificed in order to generate 'coloured' television.

This effectively works by generating two separate signals, the brightness or luminance, and the colour or chrominance. In order to transmit these signals they are combined, using different frequency carriers into the television signal. It is a well known fact that the human eye is particularly sensitive to changes in brightness. Consequently when it became necessary to retain most of the TV signal as B&W, or pure luminance, the loss in quality was sufficiently marginal to attract commercial use. Intuitively we can think of the TV signal carrying a gross signal (the luminance) which can easily be decoded by black and white sets, and finer detail (which contains the colour) which only higher quality colour sets need to examine.

This is a somewhat simplified view of the actual process, but illustrates that in any TV ready image the most important component is its luminance. A useful rule of thumb is that twice as much resolution is reserved for luminance information as for colour. The practical result of this is that a dark image, containing subtle colour changes, will appear extremely dark on a TV, and contain little detail.

The representation of colour is an involved topic, and a large variety of materials are available for the interested student [6][7]. Given the restricted bandwidth made available for colour storage the engineers were forced to concentrate the colour space sampling on regions of particular interest. So far we have represented televisions development in straightforward technical advances, but unfortunately the situation is muddied somewhat once we consider colour. Several competing designs were developed, though two are of particular interest.

The PAL system, which was developed for European users, and which most UK users are familiar with, concentrates its colour representation in the region which matches human skin tones. It is incapable of representing bright colours well, and must rely on luminance to provide this information. NTSC, the American system, instead concentrates its efforts on bold primaries. This system also uses a more involved mechanism for representing colour, and so any UK user must be aware that any PAL material he or she produces cannot be converted absolutely accurately. There are other, more technical, differences but the alternative colour philosophies are likely to have the most impact on the casual user.

The difficulties of colour representation have a marked impact on the recording of computer generated imagery. The PAL system cannot represent saturated colours at all, and most hardware would produce unpleasant 'shimmering' results when presented with such images. This means that the user must ensure that his or her images contain only pastel colours. This initially seems limiting ("You mean I can't record red?") but remember that although your images may appear washed out on your computer screen, these colours are artificially brightened on the TV image, so light pinks may well appear to be vibrant red!

2.2 Video Recording Standards

As previously mentioned many video recording systems reduce still further the bandwidth available to record the TV image. Indeed for many years no TV output was recorded, as the available media quality was felt to be too poor to bother with!

Most video standards are analogue, which means that the recording process causes a reduction in quality, as does the play-back. Furthermore it is extremely easy to damage the quality of recordings by mishandling or playback in low quality or dirty equipment. This is why it is vitally important to keep a high quality master of any footage created, of which only copies are repeatedly used.

The video standards in widespread use *today* are, in order of increasing quality:

VHS: The lowest quality tape , worst picture, poorest copying, but at the best price!. Although the Video Home System is often regarded with derision by the professional television industry, it produces perfectly acceptable pictures for domestic use, in a cost-effective fashion. Hence it became one of the few industry standards. However, there are drawbacks to using VHS as a CG recording medium. The VHS machine does not record the full resolution of the video signal and offers poor resilience to copying. As everyone has access to a VHS machine it is extremely likely that most CG material will eventually be copied to VHS for conference use.

S-VHS: Being an enhanced version of VHS, its full name being (a somewhat tongue-in-cheek) Super-VHS. By using higher quality tape and higher recording frequencies (which leads to S-VHS sometimes being called a “high band” format), S-VHS is capable of recording higher resolution images, and has improved resilience to copying. Although an improvement over VHS, S-VHS cannot record the full resolution of the video signal. Even so, S-VHS is marketed as an industrial format and is quite popular as a cost effective tool for gathering research data.

Umatic: For two decades Umatic has been the workhorse of semi-professionals. It is available in two formats. Lowband Umatic gives pictures superior to VHS, but is now virtually obsolete, whilst high band Umatic offers a picture resolution which is slightly better than S-VHS. As with VHS and S-VHS neither version of Umatic can record the full resolution of a video signal. The target market for Umatic has been squeezed by both S-VHS and Betacam, and Sony are rumoured to be considering discontinuing the production of Umatic.

Betacam SP: All the formats considered so far have recorded a “composite” signal, i.e., one where the luminance and chrominance signals are combined. Betacam (in its various guises) is a “component” recording system, which keeps the two logical streams of information separate. This offers a number of advantages. As the luminance and chrominance are kept separate (assuming the VTR is fed a component signal) there are no “cross colour” artefacts seen in a high frequency pattern represented by a composite signal. The chrominance is also allocated a higher bandwidth, resulting in higher colour resolution.

1 inch: Most video media are presented as cassettes containing magnetic tape. However before cassette technology had developed sufficiently video was recorded on large reel-to-reel systems similar to computer tape. Most of these formats have disappeared but the 1 inch format (consisting, not surprisingly, of 1 inch tapes) is still in active use. The increased size of tape allows higher quality recording and is still used in studios which have yet to move to digital formats.

D1, D2, D3, D5 & Digital Betacam: The rigours of television require the top quality tape storage format. To cater for this market a number of fully digital recording media have been devised. These are currently used by most professional TV companies in the UK, and each standard caters for a different niche. All record digital encodings of the analogue signals, rather than the RGB style files the CG market is more used to. The differing standards all produced by different manufacturers, to various specs., the principal difference being how the signal is split for digitisation. D1 & D5 each digitise component signals, while D2 and D3 represent composite information.The cost of these systems is dropping, but given the impending arrival of HDTV it is debatable whether they will ever be commonly used in academia.
Note that there isn’t a D4 standard, as the number 4 is regarded as unlucky in Japan!

The optimum recording media for computer graphics is, of course, digital, but users still need to be aware of the design goals of the signal, as D* tapes are simply accurate recordings of that signal. Further information on video recording standards is available elsewhere [2].

In designing a video suite for use in computer graphics work it is important the right standards are chosen for the right job. Whilst cost is obviously of huge importance it is equally necessary to ensure that the results will be acceptable for users. To aid comparison Table 1

Format	Approx % of bandwidth recorded	Composite/ Component	Recording Technique	Major Users
VHS	45	Composite	Colour Under	Domestic
S-VHS	73	Composite	Colour Under	Education/ Industry
Lowband UMatic	50	Composite	Colour Under	Education/ Industry
Highband Umatic	77	Composite	Colour Under	Education/ Industry
One Inch	100	Composite		Broadcast archives
Betacam SP	100	Component	Y, R-Y, B-Y, CTCM	Broadcast & High end Education
D1	100	Component	Digital, CCIR 601	High end TV graphics
D2	100	Composite	Digital	Broadcast (transmission)
D3	100	Composite	Digital	Broadcast, current BBC recording format
D5	100	Component	Digital, CCIR 601	Broadcast, current Ch4 recording format
Digital Betacam	100	Component	Digital, CCIR 602, 2-1 Compression	Broadcast

Table 1 Competing Video Standards

shows the features of each recording format.

2.3 The Future

The creation, recording and maintenance of videos is an extremely involved process. We have attempted to give a short introduction to some of the features, and problems, likely to be of interest to computer graphics users. It is worth remembering that television has developed this way for very good reasons, however irritating they may seem to CG people!

Possibly the most useful textbook for interested readers is “Broadcast Television Fundamentals” [1], which is packed full of useful material on video development.

It is worth mentioning that, thanks to the growth in multimedia, and CG use in TV production, the realms of computing and video are growing progressively closer. In the future, particularly with the advent of High Definition TV (HDTV) many of the inconveniences experienced by current video/CG users will disappear!

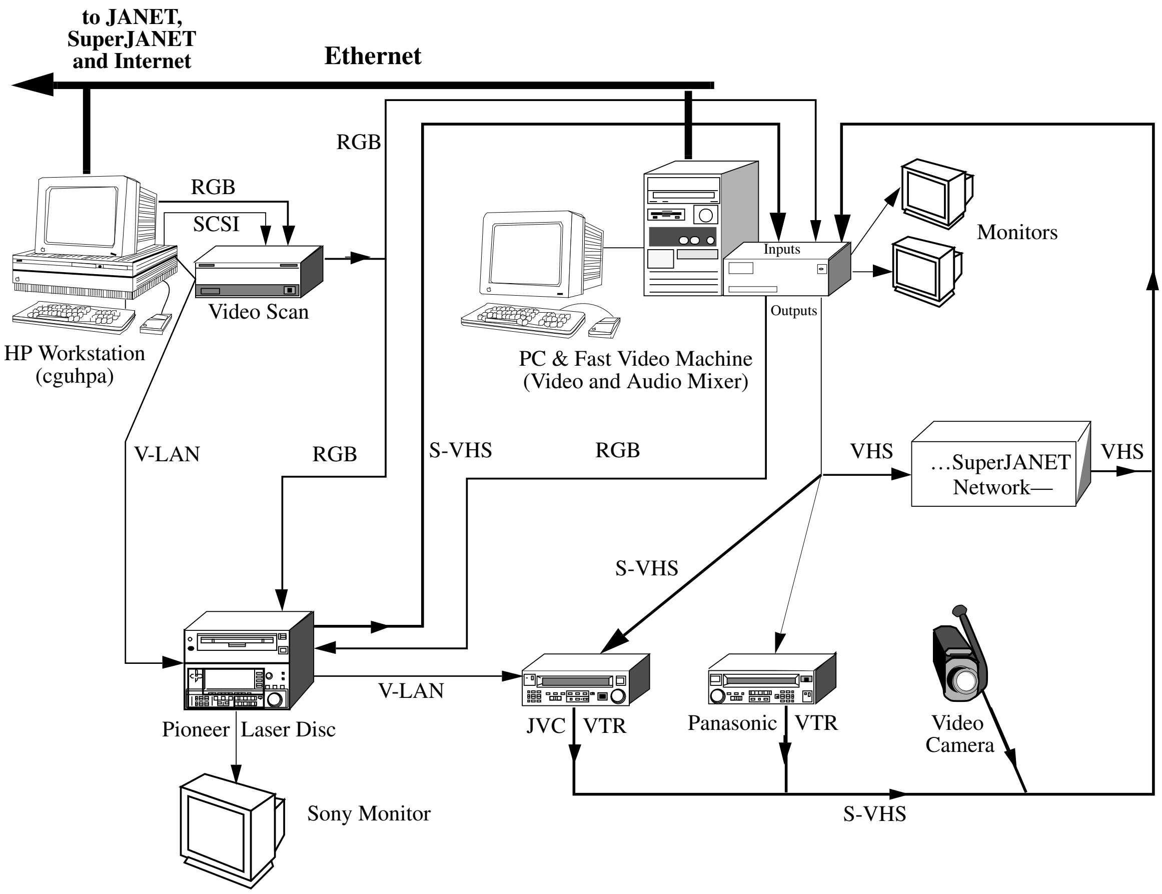

Figure 1 The Video System at Manchester

3 Video Equipment at Manchester

To provide the facilities required by academic user a collection of video equipment has been assembled in the CGU. This section contains a description of the features of this collection, and is important for two main reasons. Firstly, given the description of video technology in the previous section, it is important any user understands the quality he or she is likely to achieve. Secondly the creation of a video network is a complicated and time consuming activity. It is hoped that a description of a working system will simplify the task of repeating the process at another institution.

The CGU video equipment is primarily controlled by a Hewlett-Packard 9000/735 workstation, though a PC running a suite of video editing software is also connected to allow certain analogue effects to be achieved. The full configuration is shown in Figure 1.

The principal purpose of the system is to allow recording of images from the workstation. In this section we will concentrate on the internal workings of the equipment, though we wish to stress that a user would not need to be aware of much of this!

The principal components of the system are:

VideoScan 9100: provides much of interface hardware which allows the HP workstation to produce video-ready pictures. It is connected in series with the console monitor, to enable real-time recording of the display. In this mode the 1280x1024 picture from the console is sampled down in real-time to PAL TV resolution (768x575). It is also connected to the workstation via a SCSI port, down which digital images of TV resolution can be loaded, and output as an analogue TV signal. Software allows selection of either as source for the analogue TV signal output. Timing in video production is critical, and so we cannot use the workstation to manage much of this (otherwise the varying load on the machine would affect how well we can record videos). To produce an independent timing system the Videoscan hardware employs a network of V-LAN [4] controllers.

V-LAN-R: These small black boxes, accepts requests from the V-LAN transmitter in the VidoeScan 9100, (commands are sent via the SCSI), to control the output devices connected to the network. We will discuss this process in greater detail below.

Pioneer VDR-1000P: is a read-write laser disc system that can hold upto 30 minutes of video. This provides the primary output device, and is used to assemble video footage for eventual copying to tape.

S-VHS VTR JVC-822E: is stop-frame tape recorder i.e., frames can be laid at arbitrary positions on the tape. It achieves this by laying a timecode onto the videotape, and provides hardware to enable particular positions to be addressed on the tape. These facilities are required in even the most basic video production, and are not available on domestic machines.

S-VHS VTR Panasonic: Stop frame editing, whilst useful for video editing, is a high cost option if we only need to copy tapes. To cater for this requirement the CGU uses a high quality domestic S-VHS machine, the Panasonic.

Monitor: This monitors can be switched to view the video signal in different parts of the (analogue) system

3.1 Basic Recoding Sequence

We now discuss the needs that this equipment addresses, principally that of transferring images to tapes.

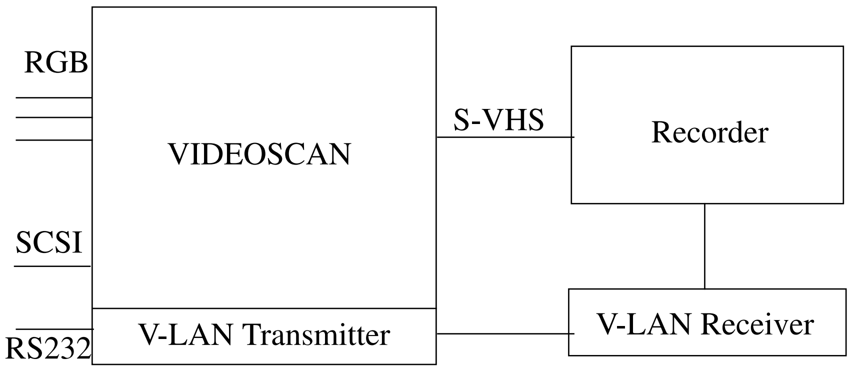

Figure 2 A simplified version of the network.

Although there is a complicated array of equipment conceptually the equipment is shown in Figure 2. The basic sequence to record one or more identical frames for n/25 seconds to video tape operates as follows.:

- Place the image in Videoscan frame buffer, using the workstations SCSI bus.
- Enable output from the frame buffer (disabling real-time mode)
- Rewind tape to pre-roll point. The format of signal all videotapes record is reliant on having

the tape heads playing at the correct speed. Consequently in order to record a single image we need to rewind the tape, and start playing. This is called the pre-roll, and with tapes is normally set to 7 seconds. Any less and we cannot guarantee that the machine will have reached the correct speed.

- Ask the V-LAN network to manage a simple edit. This is achieved by downloading a simple program to the VLAN boxes, the details of which we will discuss further below. Once this program begins executing the timing becomes extremely critical (we will probably be only recording a single frame of video).
- The V-LAN network starts the tape playing.
- The V-LAN network, at exactly the right time, asks the Videoscan to send the image in its buffer down the video cable to the VTR.
- The VTR records this frame, then slows the tape down to a halt (commonly called the post-roll).

Typically this is about 40 seconds + n/25 seconds on the JVC VTR. The hard work of synchronising the tape head with the start of scan of the analogue TV signal is handled by the V-LAN controllers. They accept an ASCII control sequence (a program). A simple V-LAN program is shown in Table 2. Again we would like to emphasise that the user does not need to

Sequence to V-LAN Tx	Meaning
`ND1`	Select the Node 1 (JVC, Node 4 is Pioneer)
`TSV`	Select the tracks (V for video, A for Audio)
`PR 7:00`	Select the pre-roll time 7 seconds
`SI 00:00:16:00`	Set the IN point. Place where frames are written
`SD 1`	Set the duration (1 frame)
`PF`	Perform the edit

Table 2 Basic V-LAN Sequence

interact with the equipment at this level of complexity!

This process is workable, but extremely slow. In order to produce any significant video footage it is necessary to leave this process working over night! To address these problems most video installations use a half way frame store, which can accept images quickly, and can transfer its contents to the tape in batch.

Some installations use digital disk recorders (effectively banks of high speed computer disks), but the CGU employs a Pioneer video disk recorder, which besides being faster to use than many digital disks (as it is not reliant on transferring images over Ethernet), offers some useful editing facilities. The video disk contains 30 minutes of Betacam SP quality footage, and operates, from the V-LANs point of view, as a very fast VTR which requires only 3 seconds of pre-roll. The Pioneer employs two read/write heads, and so can perform instantaneous cuts between video on one disk, or can be used to record and play at the same time (it is commonly used by TV companies for instant playback at sporting events).

The video disk is normally used in the CGU as a temporary frame store. A user builds up footage on the disk, and then copies longer sections onto tape for archive. In many cases, particularly in research settings, the ability to play long sequences of video from a disk has allowed users to fix problems with their images before committing to tape.

4 The Video Production Software

4.1 Software Overview

Having described both the fundamentals of video, and the equipment available to UK academic users it is now helpful to describe the process of producing videos. This section outlines the process used *internally* by the CGU software, and is intended both as background information and to supplement the features made available through the higher level interfaces described in the following section.

The most important part of the software is a library (`libvpr`) of C routines that perform many of the functions necessary, from controlling the hardware, to providing software support for the image processing that must be undertaken to prepare the images for the video. Using `libvpr` are a number of components, the structure of which is shown in Figure 3.

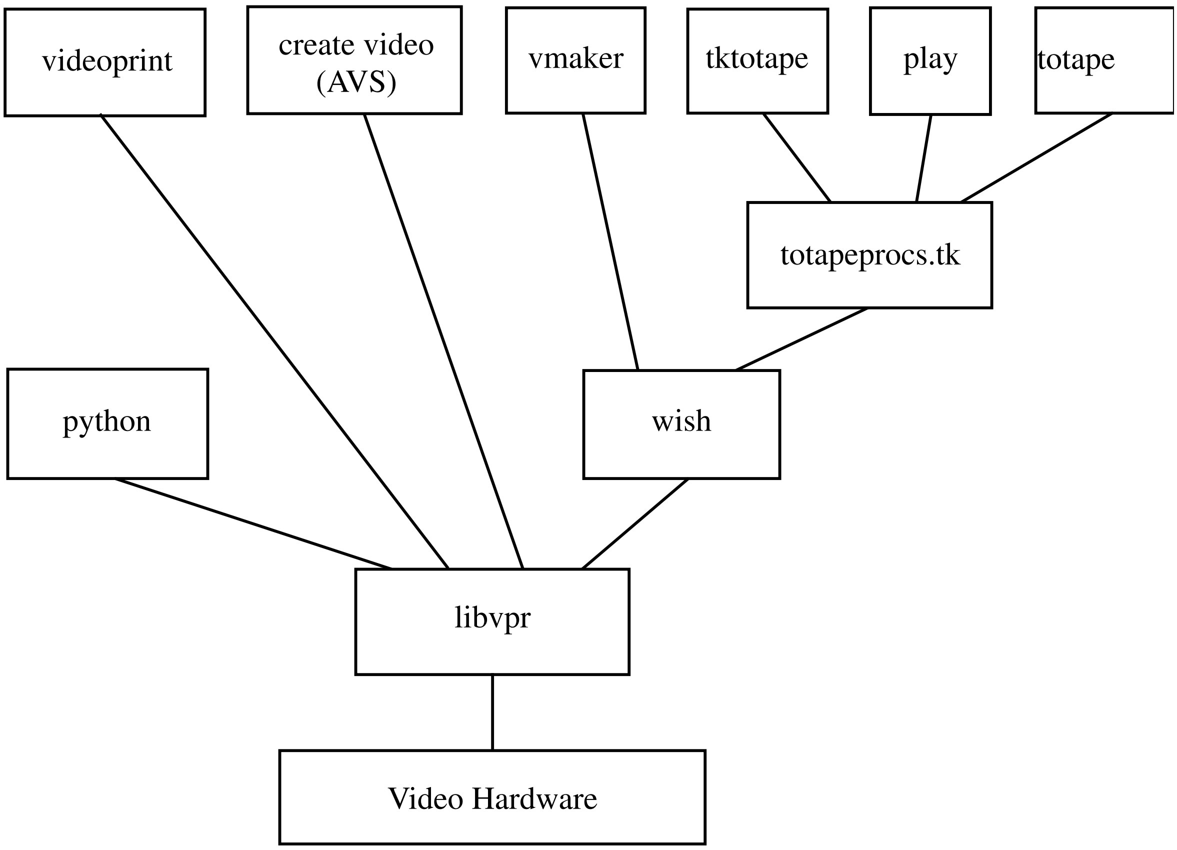

Figure 3 The software structure.

The main purpose if `libvpr` is to take a sequence of images (including in this definition PostScript), prepared by the user, and output them to either the Pioneer laser disc or the JVC VTR (or both). In addition it can:

- generate a number of ‘in-between’ frames, to create scrolling, rolling, fading and other effects.
- can generate linearly interpolated intermediate frames, to produce the effect of smooth motion.
- can generate a digital morphing sequence between two images.
- resize, through bilinear interpolation, the image to PAL resolution.

- convolute each image, in order to help a user remove high frequency areas from his or her-image.
- gamma correct images
- snap-shot the images, including the internally generated images, to TIFF files
- generate an MPEG file of all or part of a sequence.
- display the images, including the internally generated images, using `xv`.

The various software components (python, wish, `videoprint` , AVS, and `vmaker`) all access the library `libvpr` . So although they may have slightly different interface they all call the C library routines described below. We discuss these application components in section 5. Conceptually the input, from the software components point of view, is always a script of instructions.

4.2 The Video Processing Pipeline

The production of images which are suitable for recording to video is an involved process. To help a user produce acceptable results the CGU has constructed a software video processing pipeline, through which his or her images pass. By asking the various application components to enable certain features it is possible to avoid the more common problems.

After an image is read it passes through the following stages (if they are enabled):

- Convert to internal format
- Resize or reposition image to Screen Resolution * underscan %
- Generate intermediate frames (transitions)
- For each intermediate image:
  - Convolve each frame with the filter
  - Gamma correct
  - Hot colour correction
  - Write the frame to the VideoScan, Pioneer or JVC.
  - Send an image to `xv`
  - Snapshot (grab) VideoScan image to disk
  - Send image to MPEG generator

This, then, if the fundamental internal video pipeline. Sections 4.2.1 - 4.2.8 describe these steps in greater detail.

4.2.1 Supported Image Formats

Internally the software uses the Virtual Frame Buffer (VFB) of San Diego Image Toolkit. In addition interfaces to the following image format libraries are supported: CGM, ImageMagick, MPEG. Thus we have been able to support a great variety of formats quickly and easily. The full list is supplied in Appendix A. These include common image formats such as PostScript, GIF, JPEG, TIFF. PBM.

4.2.2 Repositioning & Underscan

Sadly not everybody creates images that are TV resolution (768x575) so the first thing the software does is if the image is smaller that a TV frame it repositions it with a TV frame (centred, left, right, top or bottom aligned). If the image is bigger than a TV frame then it is either cropped or resized using bi-linear interpolation. The pipeline also provides facilities to scale the images to compensate for the reduced size of the ‘safe area’ discussed in section 2.1 , which we term the underscan,

4.2.3 Transitions

Although the basic assumption of this facility is that the user has generated sufficient frames to give the appearance of smooth motion this may not always the case. This is particularly true of the captions, where a word processor has been used to create a number of pages. The software will produce a number of intermediate frames given the transition type and some parameters. The primary transitions supported are:

Simple: The new frame simply replaces the one already there. Ideal for a sequence of images which are very similar, such as rotating or moving objects.

Interpolation: generates a number of intermediate frames. Each pixel in the intermediate frames is linearly interpolated between the *old* and *new* frames.

In addition the pipeline provides support from some simple digital video effects (DVE's) for effects such as scrolling & panning. Should the user so desire the software provides hooks for additional effects to be constructed.

4.2.4 Filtering Images

As we have mentioned the video signal is incapable of representing certain image features, such as adjacent black and white pixels. If the user cannot prevent this in his or her software, the pipeline allows the application of a wide variety of convolution filters (50 in all) to remove the worst effects of such a problem.

4.2.5 Gamma Correction

One of the assumptions commonly made about RGB is that colours are linearly additive. In other words, if you have twice the red value, say, you get twice as much red light. However, a linear increase in the voltage applied to the guns does not produce a linear increase in luminance. The light produced by a phosphor is proportional to the electron beam *power*, rather than the gun voltage.

$$Power = voltage \times current$$

$$Current \propto \text{grid voltage} \;\; 1.5$$

So, in order to produce images which can be displayed on a particular monitor we must gamma correct our pictures. In contrast to general computer graphics practice, where gamma correction is the last step before display, the RGB signals which are to be recorded on video are gamma corrected before encoding. The alternative would require gamma correction circuitry in every replay monitor and, if the signal were to be broadcast, in every TV set. It is easier and more cost effective to produce a single stable, closely tracking corrector in the studio. Because of this, it is important that computer graphics images which are to be recorded on video should first be gamma corrected. While any arbitrary transform can be applied to a computer graphic image, the standards for video assume that the input signals are coming from a TV camera and that the gamma correction will be performed by an electronic circuit. The graph of a pure power law meets the origin vertically, which corresponds to infinite gain in a circuit. This is clearly impossible to attain, and would give severe problems with noise amplification, so gamma correction in a video context uses a modified power law with a linear section near the origin. To preserve compatibility, the same transformation should be used for computer graphics that are to be stored on video. The formulae given below are defined in CCIR Recommendation 709, the international standard for high definition television (HDTV). Older video standards have less precise definitions for gamma correction, ignoring the infinite gain problem. CCIR Recommendation 709 formalises the range of similar values used in modern studio equipment, and these equations are suitable for use in computer graphics. For RGB values in the range 0.018 to 1.0:

$$R\gamma = 1.099\, R^{0.45} - 0.099$$

$$G\gamma = 1.099\, G^{0.45} - 0.099$$

$$B_\gamma = 1.099\,B^{0.45} - 0.099$$

For values in the range 0.0 to 0.017, the slope of the curve is limited to 4.5:

$$R_\gamma = 4.5\,R$$

$$G_\gamma = 4.5\,G$$

$$B_\gamma = 4.5\,B$$

4.2.6 Hot colour correction

We have already mentioned how the PAL TV signal is incapable of representing as wide a range of colours as we would like. Whilst it is advisable for users to use only those colours which can be encoded onto video, the pipeline provides some facilities for removing some of the offending, or hot, colours. It achieves this by scaling all the colours in a particular image by a sufficient amount to ensure that the worst colours fall within the acceptable range for PAL encoding.

4.2.7 Writing and Snap-shotting the frame

The frame is automatically written to the VideoScan framebuffer, though, this can be disabled for testing or remote use purposes. If `xv` mode is enabled the image is sent to an `xv` process, which of course could be running at another site. These intermediate images can also be stored to disc, in one of the SDSC formats (e.g., PostScript or TIFF). Usually the frame number is encoded in the image name, which will aid any further video image processing.

This feature (with some re-cabling) also allows the facility to be used to digitize images stored on videotape or the Pioneer.

4.2.8 Making MPEG files

We have modified the public domain MPEG encoder to accept VFB rather than a list of files, so an MPEG movie can be made, at a different resolution to TV size if required, of all or part of a sequence. This facility allows a remote user to verify that the video they wish to produce is correct, and may be useful as an output medium itself in the future.

5 Application Environments

Whilst the internal workings of the video software are of some interest, the application interface which a user works with is of more importance. To encourage the use of the National Video Facility by as wide a range of users as possible a collection of higher level interfaces have been developed.

Internally each tool employs the features of the video pipeline, but is targeted to the needs of a particular category of users. The most primitive of these interfaces, a C library, is outlined in section 5.1 and is intended for use by application developers who wish to employ direct control over the video network.

For the casual user two other principal techniques have been developed. The first concentrates on providing video facilities to users of high level languages. Section 5.4 outlines the bindings prepared for users of Tcl/tk & Python. A suite of modules have been created for AVS users, which are discussed in section 5.4. Significantly these modules can be executed *remotely*, with the images being transferred transparently across JANET/SuperJANET.

In section 5.4 we present VMaker, a GUI wrapper to the video production process.

5.1 The C Function Library

Application developers may find it more convenient to control the video system directly, perhaps to reduce the number of image frames which need to be created. Here we outline the most important portions of the API, which are used internally by the other interfaces.

`boolean video_open:` initializes the hardware and software.

`video_dofile:` takes a file name as aprameter and each line of that file is submitted to `video_doline(char *string)` accepts a string containing commands. The commands fall loosely into two classes; those that set variables in the internal state, either implicitly or explicitly, and which will effect video production, and those which supply filenames which are processed through the video pipeline.

The use of strings as the major interface to the system makes it very easy to add new features without compromising existing software.

`video_close` waits for any outstanding edits to complete then closes down the hardware and software.

There are 4 functions provided to interrogate the internal state, find the status of any edits going on, and convert timecodes to frame numbers and vice-versa: `video_status`, `video_query, video_qtimecode, video_convert.`

5.2 Specialist Languages

The development of rapid prototyping environments, such as Tcl/TK and Python, have dramatically improved the ease by which many applications have been created. In recognition of the fact that many users may have experience with these tools we present two high level interfaces from these languages to the video equipment.

The Tcl/TK [3] binding includes a graphical front end to the video production process (VMaker), and as well as serving as a useful tool in its own right, illustrates how high level user applications may be constructed.

The Python [5] environment adopts an object oriented abstraction of the video, and serves as an example of how other languages may be employed in video production.

A feature of the system is that the interfaces are very similar and we present a simple example of how the same set of commands to `libvpr` would be executed in the appropriate language. Suppose a script file (sent to `videoprint`) contains the following 3 lines:

```
set exposure 3
cflace00.rle
interpolate 25 8
```

Then a C program intended to achieve the same effect would contain the following 3 function calls:

```
video_doline("set exposures 3");
video_doline("cflace00.rle");
video_doline("interpolate 25 8");
```

or if the 3 lines were in a file

```
video_dofile ('xxx.film');
```

In Tcl/TK we would use:

```
video set exposures 3
video cflace00.rle
video interpolate 25 8
```

While in Python the sequence would be:

```
video.doline("set exposures 3");
video.doline("cflace00.rle");
video.doline("interpolate 25 8");
```

A longer script file is presented in Appendix C.

5.3 The AVS Interface

Much of the UK scientific academic community is currently using AVS as its visualisation environment, and the CGU has developed several modules which provide video facilities to the AVS user. Particular attention has been paid to the ease by which the video equipment can be controlled remotely using AVS. For convenience of AVS users some of the modules that are available within `libvpr` have been recoded to work directly with AVS images. The main modules available are:

Create Video: sends an AVS image to `libvpr`. This module must execute on the relevant machine at Manchester.
Camera Control: manipulate the camera in the geometry viewer.
Create MPEG: AVS image to an MPEG file.
Image Crop: Crops images (to PAL resolution)
Image Interpolate: outputs a series of images fading from the last image to the new image.
Hot Colours: Corrects an image so that it's composite chroma limit is below 0.53 and it's total limit (Y + C) is less than 1.2 .
Gamma Correct: gamma corrects an AVS image.

Provided the user is aware of the need to employ these modules correctly (to compensate for a poor choice of colour palette for example), these modules make the video production process trivial. In its simplest form only one module need be added to an AVS network. It must of course execute on the CGU workstation with he SCSI interface, but as AVS provides the necessary remote access facilities, there is now reason (other tan network bandwidth) why the main AVS network should not execute on the user's host machine.

5.4 VMaker

So far we have presented the interface to the video system as a collection of API's. However it is not necessary for users of the system to employ such primitive interfaces. To demonstrate how more pleasant video editing software can be provided, and to serve as a testbed for research ideas, we have constructed a Tcl/Tk user-friendly wrapper to the system.

VideoMaker's (VMaker) approach is as general as possible, in that you can use it to make a video of any arbitrary sequence of images stored on the computer. You have a free choice of methods to produce the images, but the content of the images is outside the scope of VMaker, it is up to the user to produce title pages, animations and so on. What VMaker does is to allow you to assemble images in a sequence, edit the structure of the sequence, use video effects to transform between images, preview the results and finally record the video.

It was constructed using Tcl/TK and the video extensions. There are two novel features to VMaker (shown in Figure 4)

Firstly it creates a script file which is a plain text file, that can be edited with your favourite text editor, and then can be executed from within VMaker OR executes as shell (TCL/TK) script

Secondly both the timeline and the script file are displayed and are updated continuously from not only each other but also from the parameter specification part of the main display. This feature is shown in Figure 5.

Note that VMaker, besides serving as a useful application in its own right, serves to illustrate the ease by which pleasant interfaces to the video synthesis process may be constructed, provided the lower level design is sufficiently clear.

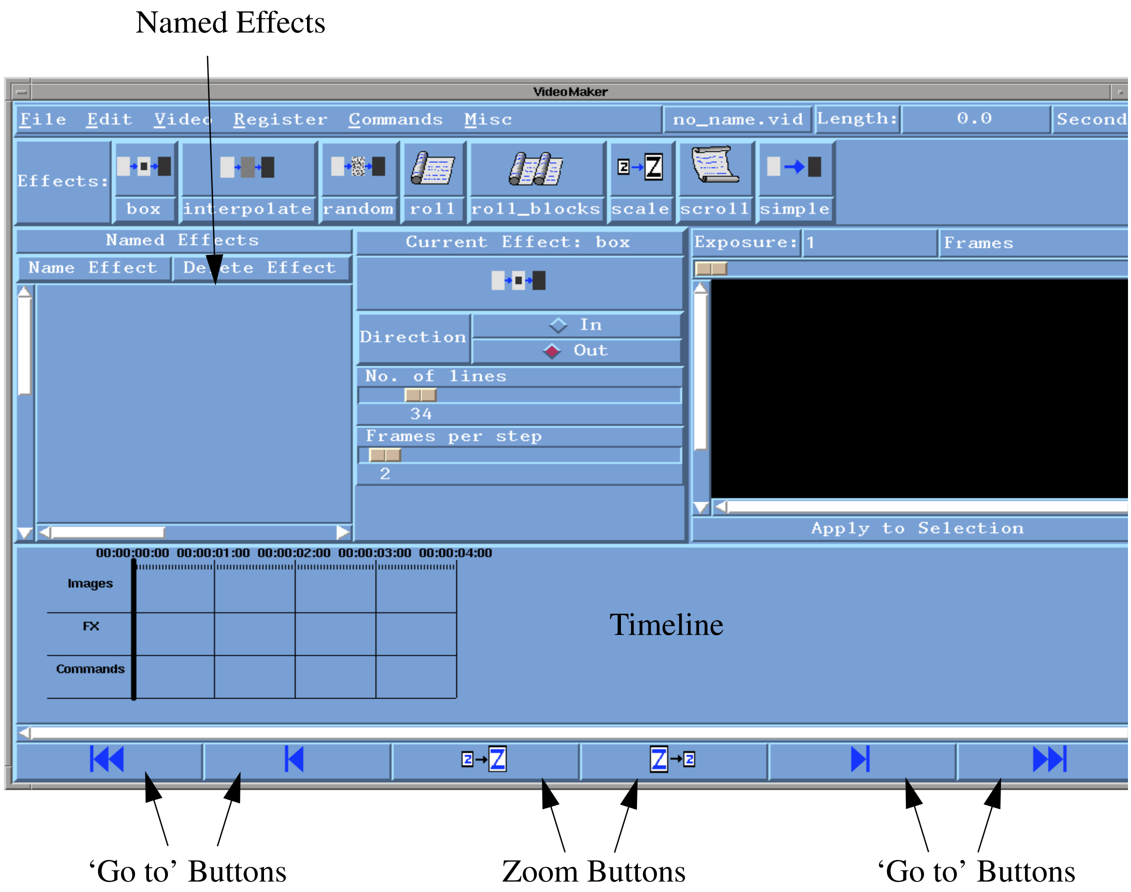

Figure 4 The VMaker Interface

6 Conclusions

We have described the design and construction of the National Videotape Facility, which is funded to assist UK academics in the production of video footage for research or teaching purposes. In presenting this system we intend the architecture to serve as a useful indication of the sorts of problems and features any computer/video installation must address.

It is important to emphasise that the difficulties of video production must be split into 2 distinct areas:

- The construction of the video hardware network. Few video suppliers are used to working with computing users, and so the process of determining exactly what hardware is required to achieve a particular result is needlessly complicated. We hope that this paper has gone some way to demystifying the process!
- The production of video-ready images. Due to the development history of TV and video it is difficult to record CG images without considerable effort. We wish to reinforce the statement that fully half the difficulties of video production must be tackled by manipulating CG images in software, as without this effort no video footage is likely to be acceptable!

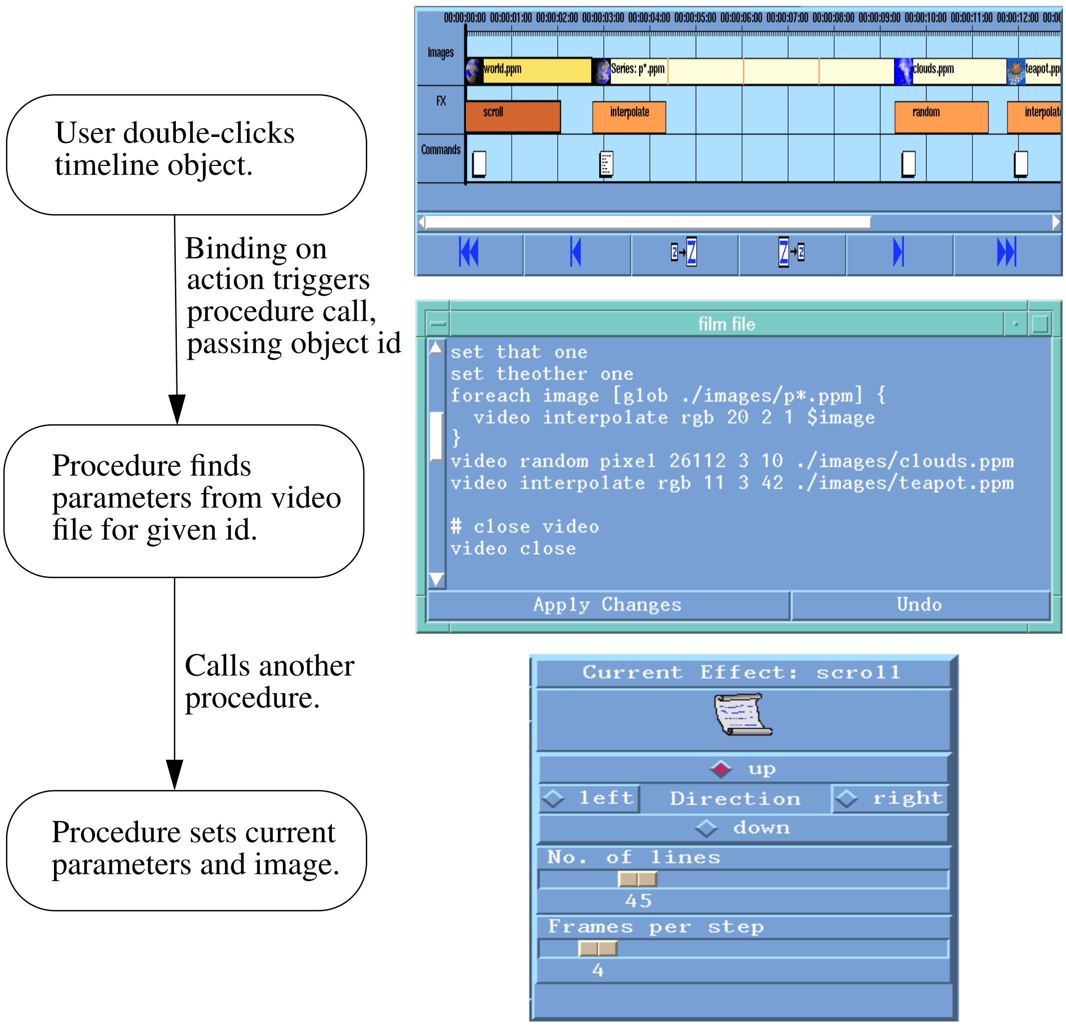

Figure 5 The double interface.

Acknowledgments

A large vote of thanks goes to David Radford, of Leeds University Television, for checking over our video information, and making numerous helpful suggestions. I (MP) owe you a drink Dave!

A number of people have made major contributions to the software: Alex Knowles, Paul Seed, and the CGU staff, all of whom are thanked profusely.

We wish to draw particular attention to the help of Chris “Colour” Lilley, without whose tireless efforts to educate us on the joys of colour science we would probably still be wondering why red looked like pink!

References

[1] “Broadcast Television Fundamentals”, Michael Tancock, Pentech Press, 1991

[2] “Video Techniques”, 2nd Edition, Gordon White, Focal Press, 1988

[3] “Tcl and the TK Toolkit”, John Ousterhout, Addison-Wesley, 1994

[4] “The V-LAN Technical Reference Manual”, VideoMedia Inc., 1993

[5] “An Introduction to Python for UNIX/C Programmers”, Guido Van Rossum, Proceedings of Dutch Users Group Meeting Nov 1993)

[6] “An Introduction to Standardised Colour Measurement”, C.C. Lilley, Proceedings of Eurographics UK 1994.
[7] WWW Based Colour information. `http://info.mcc.ac.uk/CGU/ITTI/Col/colour_announce.html.` C.C. Lilley

Appendix

A Supported Image formats

Tables 3 & 4 describe the collection of image formats which the CGU software is able to accomodate.

ALPHA	AVS	BMP	CGM (RALCGM)	CMYK
PostScript	FAX Group 3	FITS	GIF	GRAY
SGI RGB	JPEG	MIFF	Photo CD	PCX
QuickDraw	PNM PGM PBM	UTAH RLE	SUN Raster	TARGA
TIFF	VICAR	VIFF - Khoros	XBM XPM	XWD
YUV				

Table 3 Images supported through Image Magick

bmp	cur	gif	hdf, df, ncsa
ico	icon, cursor, pr	taac iff mpnt	mcap pntg
pbm pgm pnm	pcx	pic picio pixar	pict
pix alias	ras	rsgi rgb	rla, rlb
utah rle	synu	tga	tiff
viff	x avs	xbm	bm
xwd x11			

Table 4 Images supported through SDSC

B A `Videoprint` script file example

To illustrate the ease by which effects and images may be combined to generate interesting footage we include a simple script file, suitable for processing by the `videoprint` application.

```
# Film 7 : Image correction
# gamma correction
#
set exposures 75
simple vase.rgb
set gamma TRUE
vase.rgb
set gamma FALSE
#
# hot colours correction
#
vase.rgb
set hot_colours TRUE
vase.rgb
#
# hot colours and gamma correcton
#
set gamma TRUE
vase.rgb
set gamma FALSE
set hot_colours FALSE
#
# convolution correction
#
vase.rgb
set convolution blur3x3.flt
set convolve TRUE
vase.rgb
#
set convolution cntr_e.flt
vase.rgb
#
set convolution lapl3.flt
vase.rgb
#
set convolution soblvert.flt
vase.rgb
#
# hot colours, gamma and convolution correction
#
set convolution lowpass1.flt
set gamma TRUE
set hot_colours TRUE
vase.rgb
#
# end
```