Adding video into the (real)time domain

RICHARD POVALL

Dartington College of Arts, Totnes, England

The use of video as a performance medium, concentrating on interactive digital video used synchronously with realtime interactive audio, is outlined. Specifically, the author's experiments with realtime video in Macromind Director and in Opcode's Max are explored, working with motion-sensing systems in performance environments. A number of experimental pieces using sonic and visual environments driven by realtime motion-sensing performance systems are described, using Max to re-map incoming sensing data and using Macromind Director to control MIDI sequences, digital sound files, and digital video in realtime. Some of the compositional challenges presented by realtime systems are also explored, particularly the compositional issues arising from the addition of video into the (real)time domain. The author's current work in the development of a CD-ROM published in 1996 is also explored. The CD-ROM attempts to provide a genuine degree of interaction significantly more sophisticated than the usual point-and-click navigation, allowing the viewer a degree of creativity in his or her interaction with the material.

1. INTRODUCTION

Video has always been fraught with difficulties as a performance medium, rigidified by the inherent limitations of any tape-based medium, time-consuming and expensive in the extreme, difficult to rework, and hidebound by the conventions of broadcast video we have all spent so many hours of our life absorbing, consciously or otherwise.

Most of those limitations are still true – except for one. Video is now a realtime, interactive, digital medium. Of course videodiscs have been available for some years, but, except for the fortunate few who had direct access to a videodisc printer/cutter, the format is still highly problematic for use in experimental, speculative, or truly interactive performance work. At a cost of £400 or so for thirty minutes of pre-recorded material, cast in stone once paid for and mastered, the interactive advantages of the videodisc are too often outweighed by its financial and mechanical disadvantages.

Digital video, of course, remains primitive, especially within performance environments where equipment needs to be reasonably portable. High-speed desktop computers with reasonable built-in and add-on availability of average quality digital video are the harbingers of a truly interactive performance video system.

2. COMPOSING WITH REALTIME SYSTEMS

The terms 'realtime system' and 'interactive system' beg a host of compositional questions. Just what do we mean by '*interactive*'? What can '*interactive compositions*' possibly be? How does interactive performance affect our sense of the composed or designed time domain, our perceptions of how time is organised and how it passes?

There are, of course, many realtime interactive computer music systems, ranging from a simple MIDI keyboard with preset sounds, to the most complex multi-CPU system such as Tod Machover's hyperinstrument systems (Machover and Chung 1989), but there is certainly a consensus amongst classical and academic composers that a realtime computer music system involves (i) a degree of immediate, usually interactive, processing, (ii) dynamic instrumentation, controlled either by realtime synthesis or through MIDI, and (iii) probably some element of performance. Crucially, these systems imply a certain level of automation or expert knowledge – a light operator in a stage booth could, with some validity, be said to be operating a 'realtime interactive system', but this is not what we mean for the purposes of this paper.

Robert Rowe (Rowe 1993) distinguishes between 'score-driven' and 'performance-driven' interactive systems. Score-driven systems are those in which predetermined, or 'scored' materials are compared with incoming performance data, placing the composer in a more or less traditional role as the final arbiter of a piece. Composers, after all, like to control final outcomes – that control being almost intrinsic to our understanding of the word 'composition'. Performance-driven systems, on the other hand, rely on interpreting typically improvisational or otherwise loosely structured performance data, attempting to recognise *classes* of data, such as repetition, speed, pitch range, and so on. I have advocated (Povall 1995) the addition of a third classification, 'environmental-

Organised Sound **1**(2): 93–106 © 1996 Cambridge University Press

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

driven' systems – in some ways synthesising Rowe's classifications, and in other ways lying entirely outside them. In an environment-driven interactive performance system, the performer is working within an algorithmic environment in which there are elements of both score-driven and performance-driven systems. The environment is listening to the performance data, which in its turn can trigger predetermined (scored) or algorithmic, or even aleatoric processes. By the same token, the performer is also reacting to the environment, placing herself into a fully interactive feedback loop. Both of the performance pieces discussed below could be described as 'environment driven'. The composer, in building the environment, is making not so much a finished 'composition' as an instrument upon which the performer plays. More on this later.

Audio hardware and software have been in an advanced state long enough for realtime composers to have assembled a body of highly evolved work exploring multiform sonic realms mediated through a wide variety of techniques and technologies. It is beyond the scope of this paper to explore this consisiderable body of work, but experimentation has continued in this area almost since the first electronic instruments were devised. As these technologies have become more complex and powerful, their potential as performance-based systems has increased.

All too often, however, the role of the composer is rarely questioned within interactive systems. Certainly, within score-driven interactive pieces, the composer is at the top of the creative hierarchy, making a detailed score controlled by the performer's playing (in the sense that the way in which the score is reproduced responds to the interpretative gestures of the musical performer). The score, nevertheless, is performed with a high degree of repeatability. In performance-driven scores, the composer will often cede certain sections of the piece to the performer. Working with environment-driven systems takes this one step further: the composer does not create improvisational sections, but environments in which the performer plays/moves/gestures/speaks. The performer, as I mentioned above, is essentially playing an instrument, perhaps, to borrow a phrase from Machover, a different kind of 'hyperinstrument' – not based upon a physical instrument, but upon a virtual environment. Because this instrument can now include visual and sonic elements, it becomes possible to deal with notions of virtual systems in a much more organic sense – not necessarily dealing with virtual realities (although this is certainly possible), but working with virtuality as a fluid medium controlled entirely in realtime by performative gesture.

These approaches call for new ways of making work – new compositional methodologies – which I have discussed elsewhere (Povall 1994, 1995). The devising process for this kind of interactive work is unconventional, and can often be uncomfortable for the composer, and his/her creative collaborators. The compositional model moves towards that of cooperatively devised theatre, in which new work is built over a long workshopping process. The environment is built over a period of months, along with performers, artists or designers. If the intended piece is a dance performance that uses motion-sensing systems, which can be environmentally temperamental, then the work is often built in the space in which it will be finally performed. I would like to propose a generic title for this kind of environment-driven work: Interactive Performance Composition.

The conventional role of the performer is also challenged. The performer becomes significantly more than an interpreter of the score, and plays an almost equal role in the making of the work. Again, many performers are uncomfortable with this, and experience suggests that the most successful work of this type uses its creators (choreographers, composers, etc.) as performers. This need not be the case, of course, but long-term relationships between performer(s) and composer(s) are essential to these activities.

3. REALTIME VIDEO: PROBLEMS AND JOYS

Video has played a significant role in performance, again, almost since its inception as an artistic medium. In a similar manner to the historic progression seen in the development of hardware used for electroacoustic composition, as video hardware improved working methodologies became increasingly complex and tape-based. Video became less and less viable as an interactive medium, disappearing almost entirely from performance paradigms. By 1970, video was rarely seen in live performance and increasingly retreated into the studio, mediated almost to stultification by tedious and budget-consuming editing systems. Video became the prisoner of its medium – sentenced to life on tape.

Video art became quite barren for some time, with many video artists imitating their commercial brothers and sisters (as audiences also became sucked in by the passive and hypnotic nature of overproduced and oversimplified broadcast video). Ironically, this happened to the greatest degree in the cradle of the electronic art/media revolution – the United States. The highly advanced video technologies of recent years, which are generally accessible only to the few, have tended to add only lightweight overlays of glitter and gloss rather than creating richer layers of complexity and depth. A handful of artists have, however, struggled to keep video a dynamic medium, preferring more often than not to work in the

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

cumbersome and somewhat grungy confines of underpowered computers. Of course I generalise and oversimplify, but these trends have become increasingly exaggerated as television has consumed ever increasing chunks of our communal psyche.

Advances in digital video technology, and the laserdisc before it, offer an alternative route. Media artists and composers have chosen to work once again with video as a realtime medium, either within interactive installations or within performance environments. The notion of recorded video (or just-captured video) as a genuinely interactive medium is a very new one with startling implications, and, for the first time, placing the moving image firmly within the (real)time domain.

Video, by definition, has always been a time-based medium, but the availability of the moving image within an interactive, performance time domain is conceptually extraordinary. Video has become a medium as fluid and flexible as sound – one that can be manipulated and shaped into formal containers and freely operated upon by performer and composer alike. A dynamic system with equal parts audio and video is a much more natural and acceptable one for a late twentieth century audience for whom the image has for some decades been culturally dominant, almost iconic. In microcosm, the personal computer epitomises this truism, and only became widely acceptable after the advent of graphic user interfaces – equal parts dynamic text and dynamic image – became the norm.

Even though the vast majority of people have never experienced it, the notion of three-dimensional (3D) virtual worlds took the world by storm earlier this decade. Again, the notion of the individual having some (albeit illusory) sense of control over his or her visual environment in totality, was unimaginably, almost erotically, empowering. It is interesting, if not crucial, to note that not only are these immersive virtual environments *someone else's creation entirely*, but that sonic representation within them are sometimes weak, with little attempt to create any sense of virtual space or even fidelity.[1] The actual experience of these immersive virtual systems, however, is generally disappointing, always anticlimactic, and they remain too primitive to attain any sense of 'reality', let alone any kind of meta-reality or speculative reality.

Using video as an equal medium, the multimedia artist is able to design new kinds of compositional and performance environments that lead us towards the creation of entire virtual stage environments, driven by on-stage performers. The remainder of this paper will explore some of my work with realtime video – technical and aesthetic successes and failures, current and speculative systems. Specifically, I will outline two entirely different kinds of realtime performance paradigms, one set in a conventional performance space, with conventional performers, the other entirely screen-based, dealing with notions of performance in an entirely different way.

4. THE TOOLS

4.1. Realtime video in Director

Realtime digital video is still quite primitive. Apple Computer's addition of QuickTime to their operating system in the mid-1980s began to allow software and hardware designers to add motion video to their products with relative ease. Taking advantage of recent affordable hardware advances for this computing platform, the multimedia composer is now able to work on a desktop computer with one or more simultaneous windows of almost realtime video synchronised to text and music. Programs such as Macromind Director[2] allow the development of interactive multimedia sessions using digital video clips, CD-quality audio clips, as well as animations, stills and animated text. That the video frame is not full screen, and that the motion is often not true realtime,[3] almost does not matter as long as this is made a positive design feature from the outset.

Director programmers tend to build projects that are interactive in name only. The majority of Director sessions (called, somewhat confusingly, 'movies'), exemplify the 'interactive' interfaces we have come to expect from 'interactive' media such as CD-ROM. The point-and-click navigational design allows the viewer to find his or her way around material only within rigidly structured and defined pathways. Viewer choice is represented only in the direction chosen as pathways branch, not in the events that take place along the pathways themselves.

Director does not have to be used this way. It is an extraordinarily complex and powerful program, with a flexible high-level scripting language easily capable of interpreting external data from any user interface, any other appropriate software, or unusual datasets such as those from a motion-sensing system, for example. It is thus very easy, for example, to build an interface allowing the motion of a dancer on a stage, mediated through an appropriate sensing system, to control not just what video we see and audio we hear, but how fast the material is played, whether

[1] As always, there are a number of exceptions to this, including the work being done by the audio group at the National Center for Supercomputing Applications at the University of Illinois, and others.

[2] The projects described in this article used version 4.0 of Macromind Director.

[3] Realtime video is throughout defined as the ability to play back video at the frame rate at which it was originally shot, e.g. 25 frames per second (fps) for PAL/SECAM video sources, or 30 fps for NTSC video sources.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

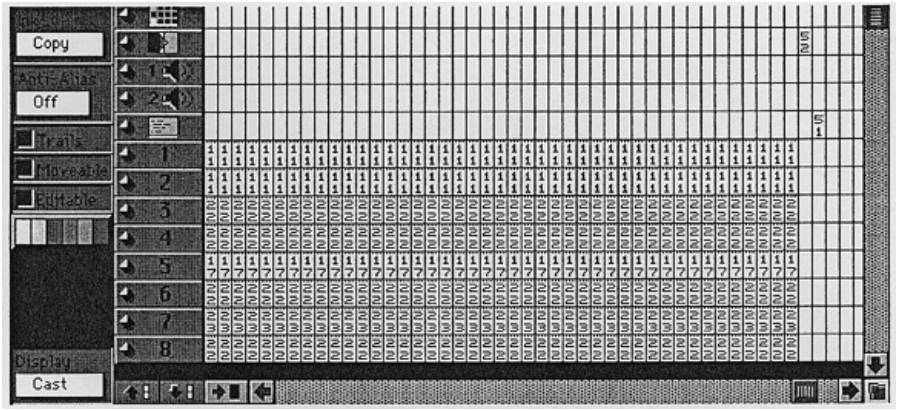


Figure 1. Score window from Director 4.0 (example from *commons:DARTMOOR*).

it is synchronous with some other element, whether it mimics physical motion or not, and numerous other parameters.

Director is highly time oriented, drawing on a *cast* (a collection of materials available to the session at any given time), placing the cast onto the *stage* (based upon the *score*) which is measured in frames. Almost every level can be controlled directly by the scripting language. Ironically, for a program designed almost exclusively to work in interactive settings, it is entirely controlled by a time-based score. The time domain is still utterly dominant within this particular framework, and only deliberate and exaggerated subversion allows the programmer to create a truly interactive session in which the time domain is controlled by the interactor (be it performer or viewer) rather than by the program itself (see figures 1 and 2).

4.2. BigEye

BigEye,[4] developed by Tom DeMeyer and Ray Edgar at STEIM, is a program designed explicitly to analyse an incoming realtime video signal and translate that analysis into MIDI data. Historically, it builds on systems developed by a number of groups on the Amiga computer, as well as on Silicon Graphics machines. Its availability on the Macintosh, together with the relative affordability and speed of video-ready Power Macintoshes, makes this program a notable advance on its predecessors.

The beauty of BigEye is the number of different ways in which it can look at an incoming signal, the number of possibilities for subdividing its virtual stage, the number of layers that can be operating simultaneously, and the ease of programming. Specific types of motion can produce simple MIDI responses (note ons/note offs, pitch changes, etc.), can send more complex MIDI data to external programs such as MAX or Director, or can work with the high-level scripting language built into the program.

Figures 3–5 show BigEye responding to incoming data in a variety of ways. For example, the program is able to watch the entire incoming video signal, or only part of that signal, and follow a particular colour (figure 3). Alternatively, it can look for difference, or change in signals – an excellent way to track motion of a single body against a static background, or to detect light changes on an otherwise static object (figure 4). The program also allows the user to split the stage into segments, either simple subdivisions, or hand-drawn masks (figure 5). The combination of all these functions provides an exceptionally flexible and powerful motion-sensing environment.

BigEye is not used directly in the case studies presented here, but was used in the development of materials ultimately used in *mouth*.

4.3. MAX

Opcode's MAX, currently in version 3.0, is a compositional programming environment designed primarily with musicians and the needs of the composer in mind – indeed, the program was originally developed at IRCAM, France's seminal computer music research centre. MAX is widely used and well known, and it is not appropriate here to attempt a detailed description of the program. Suffice it so say that MAX is an ideal environment for developing interactive pieces, largely because of its innate lack of a timebase, but also because of its amazing ease of use and resulting fast development time. Programming in MAX is practically realtime, making it possible to actually write new programs (or 'patches' as they are called) in rehearsal or workshop sessions, with performers present, without undue delays or waiting around time. MAX also handles communication with the outside world extremely well, particularly through MIDI, making it a relatively simple matter

[4] Further information about BigEye is available from the STEIM Foundation, Achergracht 19, 1017 WL Amsterdam, The Netherlands. STEIM may be contacted at steim@xs4all.nl, or point your browser to http://www.xs4all.nl/~steim/.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press



Figure 2. All windows (stage, score, transport, cast) from Director 4.0 (example from commons:DARTMOOR).

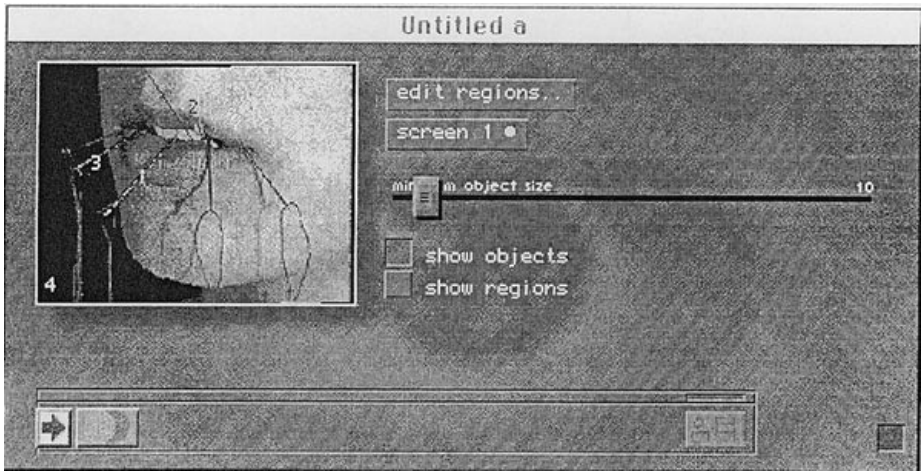


Figure 3. BigEye 1.0: tracking a colour (blue/numbered lines show tracking activity).

to interface with motion-sensing systems (see figure 6 for an example of screen shots of a MAX patch).

5. A REALTIME PERFORMANCE SYSTEM FOR AUDIO AND VIDEO

By way of specific example, I would like to present three case studies consisting of collaborative pieces made over the past few years. These pieces represent three different approaches to realtime video. *The Last Garden* uses three laserdiscs plus one channel of video on tape, as well as MIDI-controlled computer music, environmental sound, and stage lighting, together with a video-based motion-sensing working with an ensemble of four dancers; *commons:DARTMOOR* uses five ultrasonic beams to sense the motion of a

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

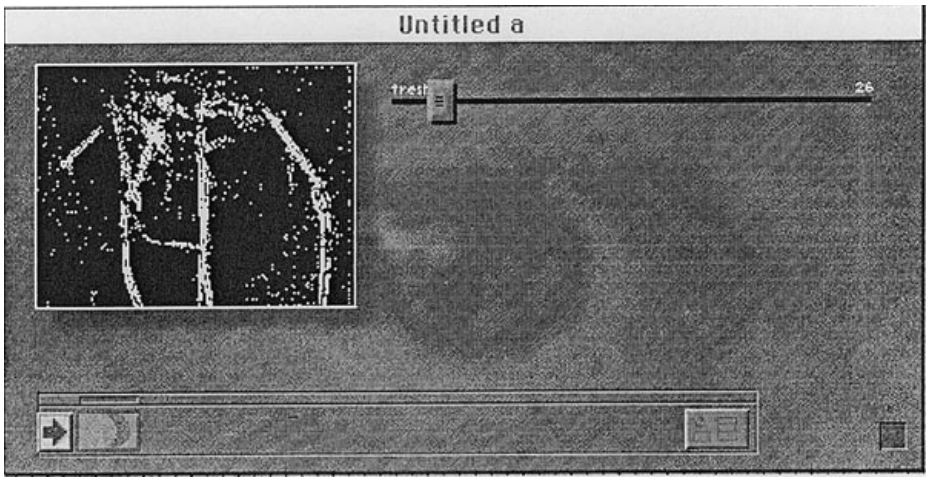


Figure 4. BigEye 1.0: tracking difference/motion.

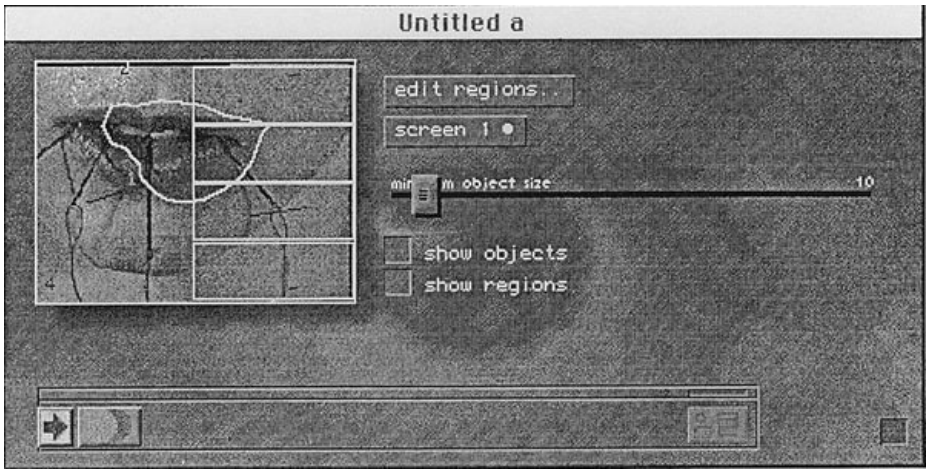


Figure 5. BigEye 1.0: splitting the stage.

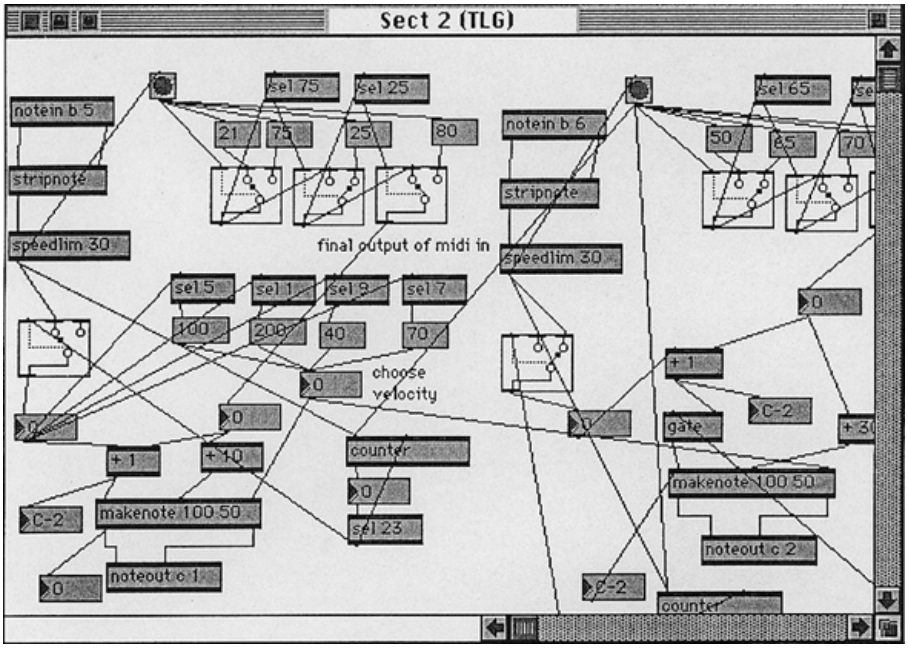


Figure 6. A MAX patch from *The Last Garden*.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

solo dancer, driving four channels of sound and controlling digital video within Director; *mouth* is a CD-ROM currently in development – a screen-based piece designed for single viewer/interactor use.

Each of these pieces was made using an interactive performance composition methodology, i.e. in workshop, where performance environments and materials are made *in situ*.

6. CASE STUDIES

6.1. *The Last Garden*

The Last Garden was made in 1992–3 at the Institute for Studies in the Arts (ISA) at Arizona State University, Tempe, and at the iEAR Studios at Rensselaer Polytechnic Institute in upstate New York. It was made in collaboration with three other artists: John Mitchell, live electronics/computer music; Kathleen Smith, choreographer; and Jeff Thompson, designer. The piece is a seventy-minute long interactive performance with an ensemble of four dancers, and uses a video-sensing system developed at ISA by John Mitchell and Robb Lovell (Lovell and Mitchell 1995). Although eventually realised with only two channels of video, the piece was originally designed as shown here, with three channels of interactive video on videodisc, an additional channel of fixed video on tape, four channels of sound, and computer-(MIDI) controlled lighting. In addition to the motion-sensing system, a number of hardware (light-based) triggers were used on the stage for the final section of the piece. Figure 7 shows the technical schematic for the piece, and figure 6 shows an example MAX patch from the piece.

Interactive video is used in a number of different ways. In a series of solos, the stage is divided by a number of trigger points, each of which triggers playback of a short video clip – an extract (up to ten seconds long) from a series of video interviews with the dancers talking about themselves. The performers are able to interrupt their choreographed solos by triggering these video statements, although the system is programmed in such a way that the particular statement triggered is a random selection. Each trigger, then, is not pre-programmed with a particular video clip, but selects one of a number of possible clips. In another section, the dancers trigger specific images, as they can trigger specific sampled sounds by, again, passing through triggers on the stage.

This was my first attempt at working with video in realtime, and the level of interactivity is primitive at best. In some sense, nothing was done that could not have been done by an operator pushing start and stop switches, but philosophically, it was important to provide the performers with a degree of control over when material was triggered, and to model the *sense* (essence) of the movement on stage in the ensemble section by triggering video clips in addition to sound samples.

6.2. *commons:DARTMOOR* – digital video in performance

commons:DARTMOOR has never been shown in final performance, but was seen as a work in progress at the 1995 Digital Creativity (CADE) Conference at the University of Brighton, England. The piece is a collaboration between the author and the choreogra-

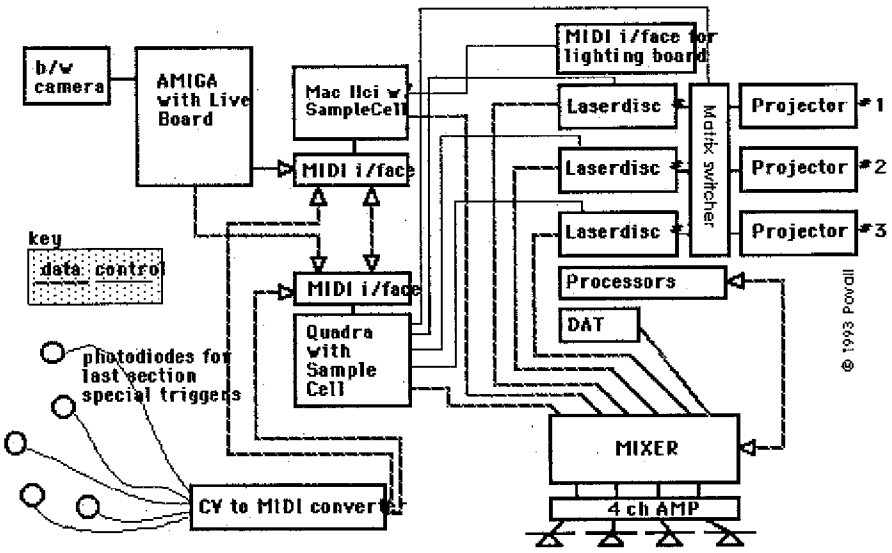


Figure 7. Technical schematic from *The Last Garden*.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

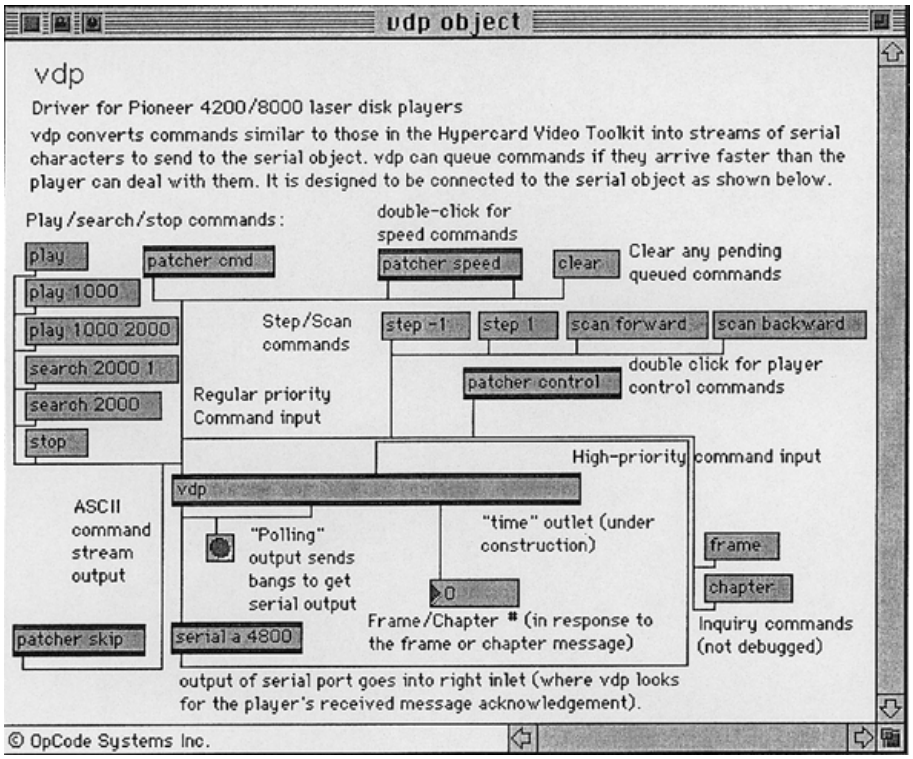


Figure 8. VDP object from MAX 2.5.2, used to control videodisc players.

pher/performer Louise Blackwell, and uses five ultrasonic sensing beams to send information about the dancer's movement to the computer systems. There are two Macintosh computers running here – a Quadra 700 as a master controller, and controlling audio, and a PowerMacintosh 8100 running digital video in Director. The processing demands of running multiple channels (windows) of digital video are heavy for these computers, and, while it is theoretically possible to run this entire system on one CPU, it is highly desirable to split the computing demands between two machines. The operational requirements of having to see the MAX window as well as running Director in the foreground also necessitate using two machines (see figures 10 and 11).

In *DARTMOOR* each of the ultrasonic beams can potentially react differently, and the performer works directly with the chosen parameters (see figure 12).

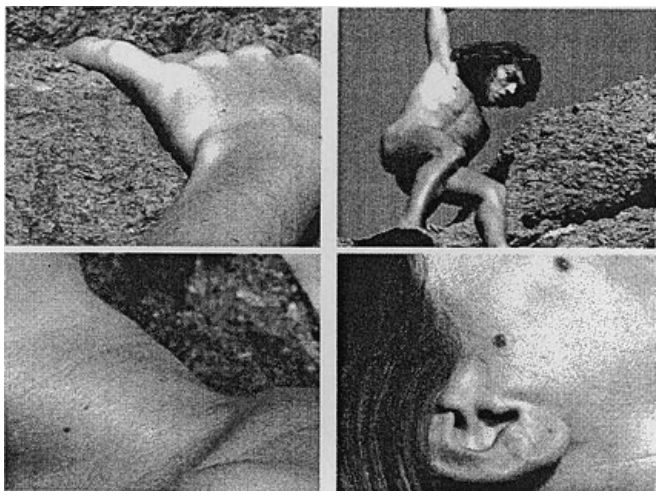


Figure 9. Images from *The Last Garden*.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

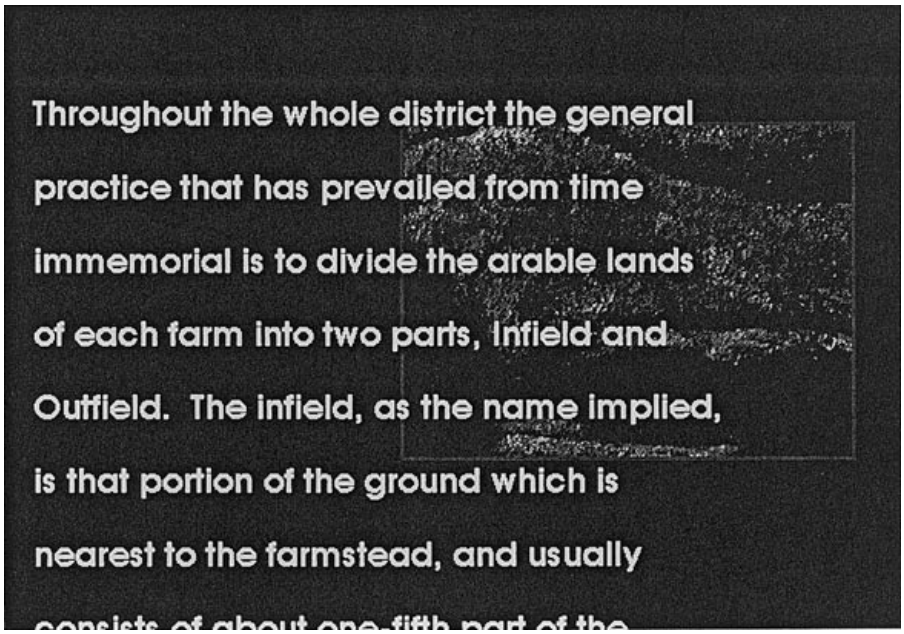


Figure 10. A screenshot from *commons:DARTMOOR* showing scrolling text.

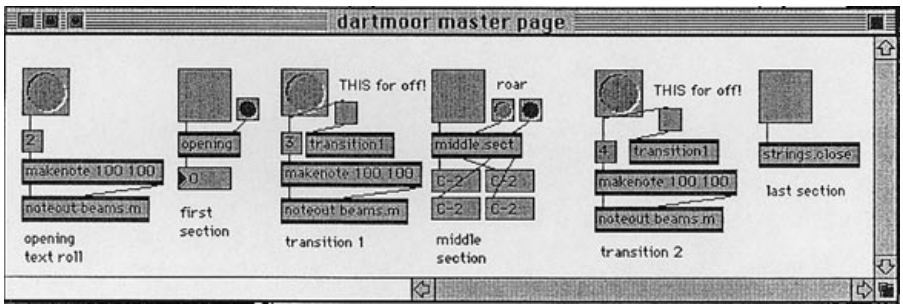


Figure 11. The master controller MAX page from *commons:DARTMOOR*. The simplicity of this design allows simple user operations during the performance, and is designed with big buttons and with as little visual clutter as possible.

This is an entirely different kind of motion sensing from the video-based system used in *The Last Garden*. Its tracking of motion is much more precise – and much more literal. Literal translation of motion is not necessarily desirable, however, and it quickly becomes apparent that close correspondence between a particular motion and a particular sound or image can be uninteresting. The ultrasonic beams are simply sensing how far an object is away from their source. A series of beams is thus able to sense how an object is moving within a space (but not with great accuracy because of the amount of spurious data coming out of the system). Instead of attempting to filter out this spurious information to create a detailed numeric picture of how the solo dancer is moving within the space defined by the beams, I use each beam in a different way: two to control pitch, one to control fine pitch or pitch bend, one to control a series of animations, and one to select timbres. The use for each beam changes in each section of the piece, and in devising the work the choreographer was able to work with these to develop different sections of movement, along with the sonic and visual environment which was also developed in coordination with the movement.

The degree of control over the images is somewhat greater than in *The Last Garden*, although still rather primitive. Because the source is now digital video on the computer rather than pre-recorded sections on videodisc, it becomes possible to add a much greater degree of flexibility. At times, individual frames are called up to create animations from video, giving the performer the ability to control speed and direction (forwards/backwards) of video clips. It is also easier to call up a wider variety of video clips because the speed of access is considerably greater than videodisc,

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

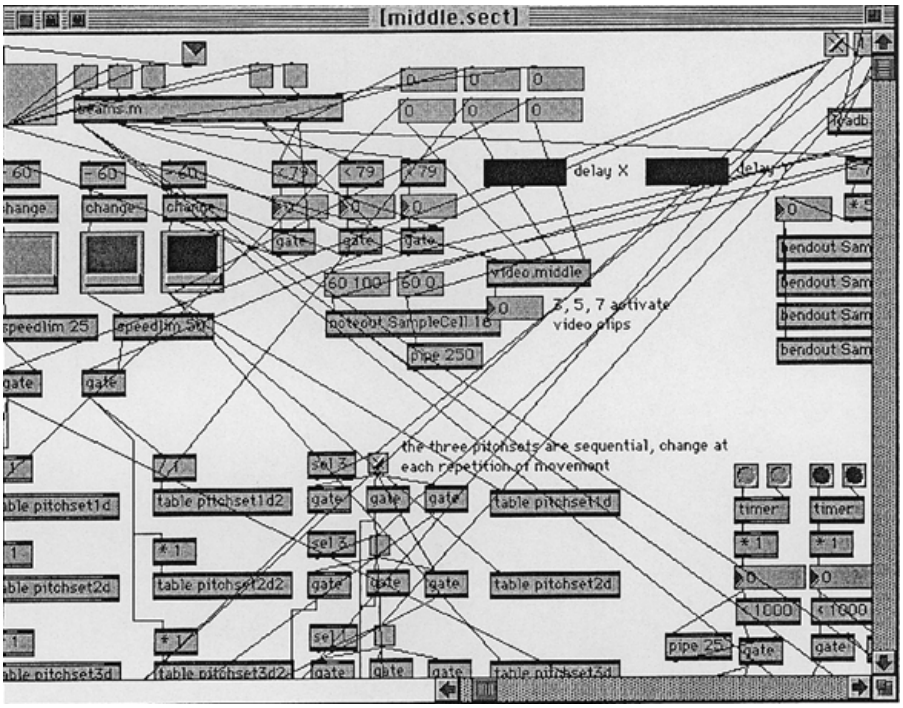


Figure 12. A MAX patch from *commons:DARTMOOR*, showing mapping of input from ultrasonic beams.

and it theoretically becomes possible to add live video processing under the control of the performer (although this was not explored here). One could also work with text in interesting ways. Animated text and sampled spoken word are now part of the vocabulary of the performance environment, and the performer is able to work with text in remarkably flexible ways. Again, this was only superficially explored in *DARTMOOR* but is at the core of the CD-ROM work in *mouth*.

6.3. *mouth* (an artists' CD-ROM) – composing for on-screen performance

mouth (working title) is an experimental artists' CD-ROM under development at the time of writing by the author and the feminist performance artist and theorist Jools Gilson–Ellis. Attempting to go beyond the conventional point-and-click navigational interactivity of the typical CD-ROM, *mouth* attempts to use an interactive performance composition model to build a unique interface allowing the interactor (a single viewer) creative access to the materials. The interactor, then, will actually be *making* the piece as he or she explores the materials, rather than simply navigating around pre-existing discrete sections. Obviously, this can never be entirely true with this type of technology and material, but there is certainly a sense in which the interactor is the performer – there is a direct analogy. Of course there are pre-recorded materials, and they are bound to present themselves with a certain sense of formalism and linearity, or at least with a likely set of coincidences. However, the compositional urge is to create a piece where this happens as little as possible, rather than a piece where this happens as much as possible – the more typical compositional model. This challenges fundamentally the ways in which the artists are able to develop thematic ideas, or even to present those ideas – particularly as the piece is overtly exploring feminist notions of the mouth and dealing very directly with physicality. Indeed, the time domain itself is of necessity deconstructed entirely.

CD-ROM as a medium has severe technical limitations, even using the fastest (quad speed) CD-ROM available. The videoCD is probably the ideal format here, because of its ability to store more than an hour of full-frame video and play it back in realtime. However, this format is far from readily available – at the time of writing this article, the special players required for videoCD are only just beginning to emerge onto the commercial market.[5]

mouth had only a six-month development time available, and is now complete. The technical limitations of the medium, and the variability of how multimedia information can be viewed on different computer platforms, and even on different CPUs within the same platform, forced a number of decisions before the project even began. *mouth* is a

[5] *mouth* is available in the UK from the author, in the US and elsewhere from Frog Peak Music (a composers' collective), Box 1052, Lebanon, NH 01766, USA, or on the internet from http://smallmedia.com/bin/fgpstart.cgi/token=856093/image=yes/desc=yes/pub=fgp/pubcall=yes/searchterm=@/track=search/.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

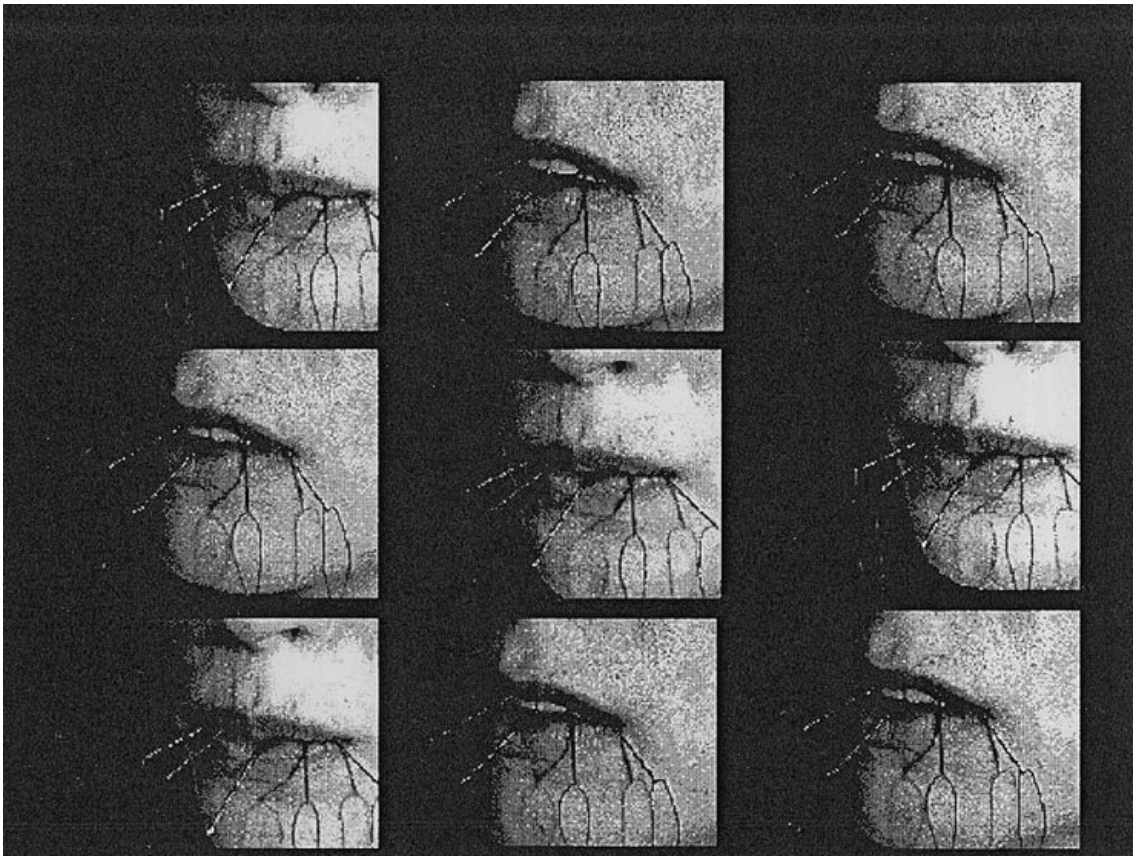


Figure 13. Screenshot from *mouth*.

Macintosh-only product – the difficulties of moving digital sound and video files from one computer platform to another being too significant to address given the short development time of the project. The finished product cannot even be viewed successfully on all Macintoshes. It is designed using 16-bit sound, a minimum requirement for any sound artist, and therefore requires a Macintosh with 16-bit sound playback capabilities. While any current generation machine, or any older machine fitted with a 16-bit sound card will do this, this still leaves out a significant portion of the installed base. Similarly, not all CPUs are fast enough to play 16-bit sound and real-time video simultaneously. The Director files are crafted in such a way as to allow maximum speed at playback and minimum load on the CPU, but the reality here is that, unless the Macintosh has at least 16 MB of RAM, 12 MB of which are allocated to Director, the CD-ROM will not run well, and the video is unlikely to approach realtime.

Having accepted these limitations, considerable effort is placed upon organising material, particularly video, so that it can run at maximum speed. Version 4.0 of Director, used for the development of *mouth*, does have significant speed improvements over its predecessors, and claims to run digital video much more efficiently. However, video windows recorded at 192×144 still have difficulty running at 25 fps[6] if there are significant additional overheads, such as animated text objects, or 16-bit sound files. As *mouth* often uses multiple video windows, the overhead becomes worse.

Several strategies have already been tried to save these speed problems. A composer used to the luxury of working with long seamless digital audio files is in for a shock! Long digital sound files will not necessarily play well when the computer is also trying to play 25 frames of video per second, and sound files begin quickly to break up. The only solution to this is to make very short sound files and preload them into RAM. Director's RAM handling is excellent, and it is capable of juggling multiple objects in and out of RAM with relative ease – assuming enough RAM is available. This trick will not work with video files,

[6] The original video was shot in PAL.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

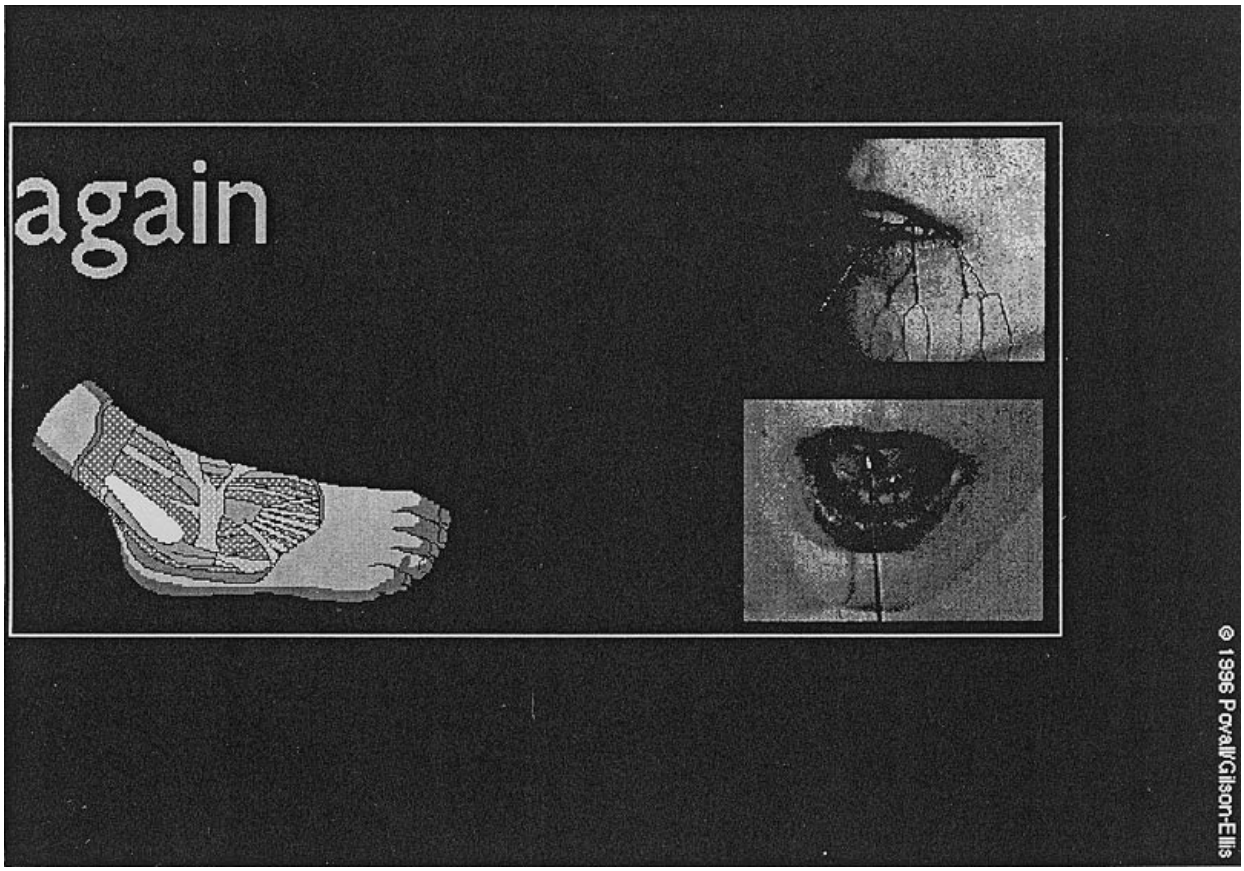


Figure 14. Screenshot from *mouth*.

however. Even the shortest video clip is usually too big to play from RAM, and other strategies for dealing with video need to be developed. One of these strategies is simply to speed up as many of the other processes as possible, not using, for example, any other kind of animation requiring fast screen redraws at the same time as video is running, using short video clips, making backgrounds simple (but elegant!) to keep as much RAM as possible free for general program overhead, all of which helps video to run smoothly. Using MIDI for sound might also be a solution, as Director supports MIDI reasonably well. While this may be ideal for performance situations, it will not, of course, work here because of issues of portability. However, it may be possible to work with QuickTime MIDI instruments – sounds stored on disk and accessed as though they were MIDI sampler sounds. This approach is yet to be tried at the time of writing.

Where possible, by far the fastest way to present digital video, however, is to convert files into PICS files. PICS is the universal animation format, and, essentially, splits a single video file into individual frames. As Director 4.0 is capable of drawing upon a cast of up to 32,000 'characters' at any given time, this option is worth pursuing, but is much heavier on RAM as it preloads its PICS objects. Video, however, essentially becomes animation, and can fly at up to quadruple speed so that the interactor is able, for example, to use the mouse or keyboard as a realtime video frame controller, allowing gestural control over the speed at which video is presented – a very significant level of creative interaction.

mouth is a feminist exploration of the mouth, and explores notions of mouth and mouth-ness from ten different perspectives. At the time of writing, these include food, hysteria/madness, articulation/utterance, singing, the mouth in relation to the body, and the mouth as a sexual object/tool. It uses video, animation, text animation, spoken text, sound textures, and music. Each section stands entirely alone, and it is unclear at this point whether certain materials will have to be copied onto a local hard drive in order to run. It is possible to ask Director to check on launch of the project what CPU is being used, which could then prompt the interactor to copy certain chunks of files onto a local hard drive before attempting to use *mouth*, or even automate that process. The entire project is highly speculative and experimental, and the artists consider it to be a first experiment rather than an attempt to create a new genre. There are of course other artists' CD-ROMs, not the least of which is Laurie Anderson's/Hsin-Chien Huang's *Puppet Motel* (Anderson and Huang 1995). Certainly the

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

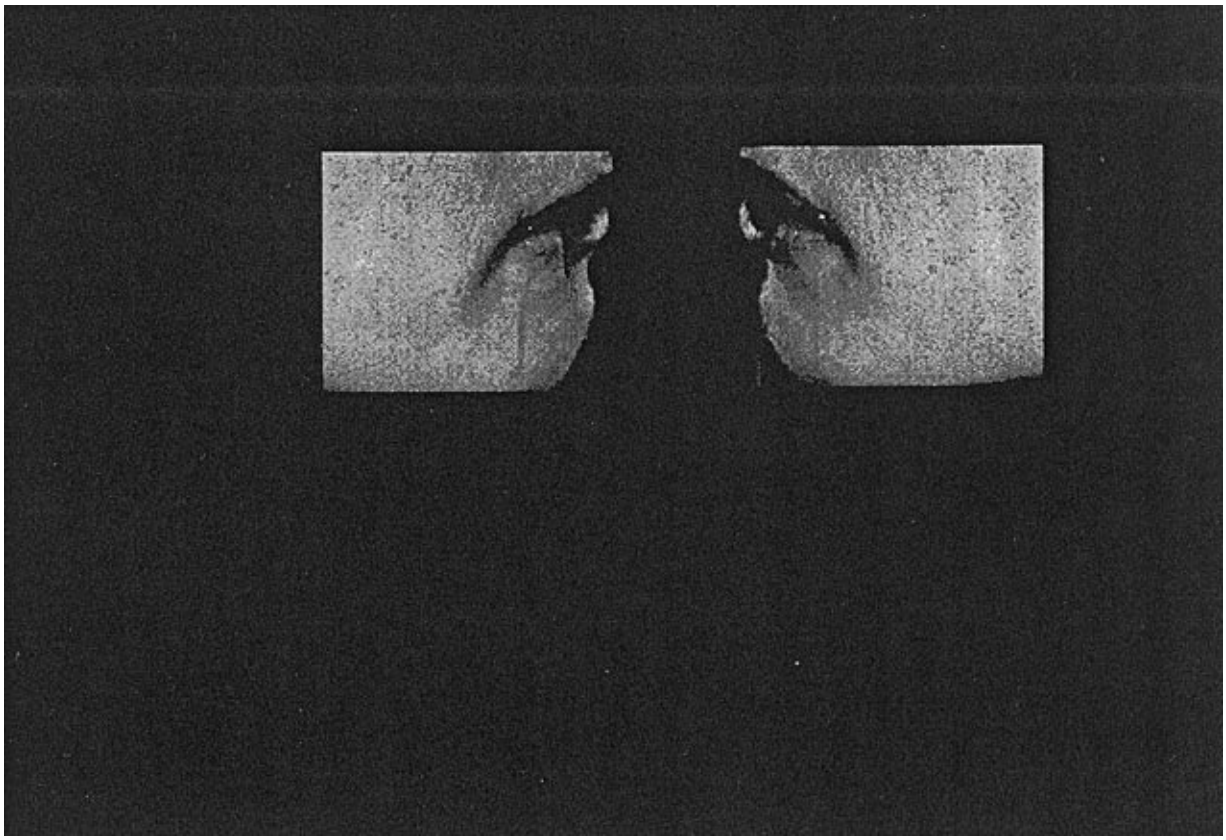


Figure 15. Screenshot from *mouth*.

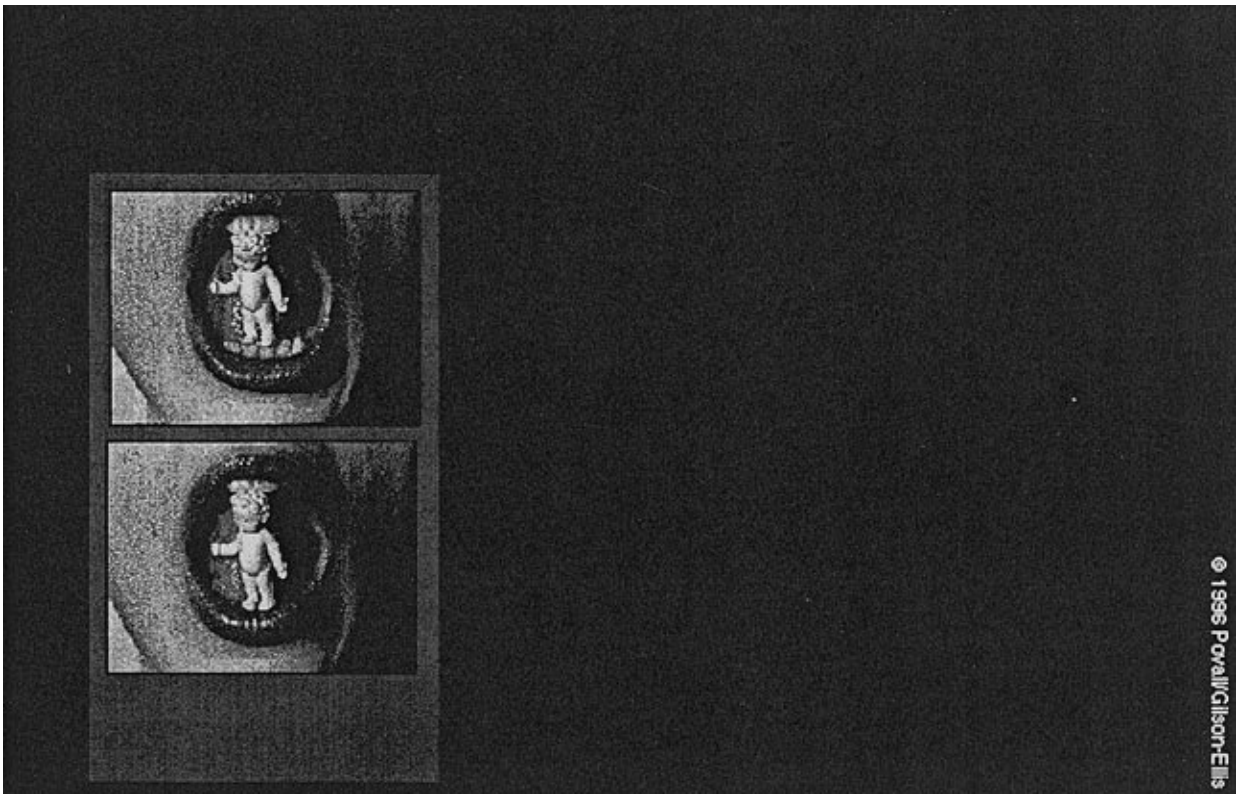


Figure 16. Screenshot from *mouth*.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press

technical limitations of the medium have a considerable impact on the kinds of materials chosen and the ways in which they are used, but an aesthetic choice from the outset was to design screens that are sparse, clean, and 'un-virtual'. By this we mean, very much in contrast to *Puppet Motel*, that no attempt is made in *mouth* to build 3D virtual worlds, largely because this is an aesthetic world both artists eschew. Indeed, to some degree, *mouth* eschews the received tenets of multimedia design altogether. There are few buttons and few arrows – this is not, after all, primarily a navigational product. Instead, plain coloured boxes on black backgrounds predominate, and little attempt is made to 'dress' the video. This simplicity also gives the interactor a more significant impact upon the visual environment. Windows appear and disappear on input from the interactor, sounds change, video material changes, plays forward, in reverse, upside down. The screen is a playpen, a personally creative space rather than a deeply designed, over-crafted artificial world. Figures 13–16 show a series of screen shots from *mouth*. These design decisions and interaction design decisions do not come from a *laissez-faire*, let-the-viewer-make-the-piece derogation of artistic responsibility, but from a genuine desire to make work that is truly interactive – the kind of work I have been attempting to make in conventional performance contexts for a number of years, some of which has been outlined earlier.

In the process of developing *mouth*, the authors were commissioned by the Arts Council of England to create a small digital work for distribution via the World Wide Web. As a result, a re-built section of the CD-ROM is available for download from http://www.ace.mdx.ac.uk/Hub/Hub.html.

7. CONCLUSIONS

For the first time, pre-recorded video has become a realtime tool, putting it above, rather than on a par with, live video, e.g. video sourced from a live camera or image processor. The ability to work with recorded video interactively, to shrink or stretch video time, to call up frames of video almost instantaneously, to, in essence, *edit on the fly* is an extraordinary addition to the interactive artist's palette. There are few who will pursue this area – many composers uncomfortable with the notion of interactive performance composition are even less comfortable working collaboratively with a video artist in the kind of intimate relationship that this kind of work engenders, almost demands. It is, however, an exciting area of research, still primitive and frustrating, but part of the inexorable march of the image – increasingly dominant in contemporary life and contemporary art.

References

Anderson, L., and Huang, H. C. 1995. *Puppet Motel*. New York: The Voyager Corp.

Lovell, Robb, E., and Mitchell, J. 1995. Using human movement to control activities in theatrical environments. In *Transcending Boundaries: Dance and Technology III*. Columbus, Ohio: Fullhouse Publishing.

Machover, T., and Chung, J. 1989. Hyperinstruments: musically intelligent and interactive performance and creativity systems. In T. Wells and D. Butler (eds.) *Proc. 1989 Int. Computer Music Conf.* San Francisco: International Computer Music Association.

Povall, R. 1994. The Last Garden: explorations in interactive performance methods. *Leonardo Music Journal* **3**. Boston, MA: MIT Press.

Povall, R. 1995. Compositional methods in interactive performance environments. In M. Leman and P. Berg (eds.) *Journal of New Music Research* **24**. Lisse, The Netherlands: Swets & Zeitlinger.

Rowe, R. 1993. *Interactive Music Systems*. Cambridge, MA: MIT Press.

https://doi.org/10.1017/S1355771896000064 Published online by Cambridge University Press