A generic model for compositional approaches to audiovisual media*

ANDY HUNT, ROSS KIRK, RICHARD ORTON and BENJI MERRISON

Music Technology Group, University of York, Heslington, York YO10 5DD, UK
E-mail: adh2@york.ac.uk prk1@york.ac.uk ro1@york.ac.uk

The challenge of composing both sound and moving image within a coherent computer-mediated framework is addressed, and some of the aesthetic issues highlighted. A conceptual model for an audiovisual delivery system is proposed, and this model acts as a guide for detailed discussion of some illustrative examples of audiovisual composition. Options for types of score generated as graphical output of the system are outlined. The need for extensive algorithmic control of compositional decisions within an interactive framework is proposed. The combination of Tabula Vigilans Audio Interactive (TVAI), an algorithmic composition language for electroacoustic music and realtime image generation, with MIDAS, a multiprocessor audiovisual system platform, is shown to have the features desired for the conceptual outline given earlier, and examples are given of work achieved using these resources. It is shown that ultimately delivery of new work may be efficiently distributed via the World Wide Web, with composers' interactive scripts delivered remotely but rendered locally by means of a user's 'rendering black box'.

1. INTRODUCTION

Music has a well-developed aesthetic and praxis derived from centuries or even millennia of cultural enterprise. The same can be said about the visual arts. The dynamic and structural form of music has resulted in an established philosophy and understanding which has aided the musical discourse between composers, performers and audience. A similar development of philosophy in the visual arts has taken place, although the growth of understanding relating to the dynamics of the moving image has been largely confined to the present century.

It is only in very recent times that the means have become available to synthesise (or 'render') image and sound concurrently within computers, to produce genuinely integrated *audiovisual* art forms. However, this degree of integration is only possible when all aspects of the composition and performance process can take place within a coherent environment or framework.

Even when such a framework is available, there remain difficult aesthetic and philosophical issues regarding the exploitation of the new art form. For example, what does it mean to 'perform' a dynamic image, linked to a concomitant soundscape? How should such an audiovisual work be organised and composed? How should the balance of responsibilities between the composer, the performer (and the audience) be arranged?

These issues are becoming urgent. As noted above, we now have, or will have very soon the technical means to realise these multiple-media compositions. Throughout the centuries artists have been quick to exploit technological development, and we must expect the same to happen with audiovisual digital art forms. The time is now right to begin at least a preliminary debate on the organisational framework for the combination of the two media.

Unfortunately, something of a Gordian knot exists in developing such a philosophy. It is difficult to develop an aesthetic until tools are available which provide the means to realise artworks within the aesthetic. However, it is also difficult to justify the effort required in developing the tools, unless the aesthetic exists which can provide a context for their use. In this paper we attempt to break this closed system by describing some simple integrated techniques which we have used for the rendering of sound and image. Even at this early stage of our work, we have been forced to address the question of 'What does it mean to compose an audiovisual work?' We therefore also describe our preliminary investigations into this issue. Whilst we would not claim a fully developed aesthetic at this stage, we hope that our work can contribute to the continuing discussion regarding such a development.

We start by recognising that there are two outputs from an audiovisual work: the image and the sound. The problem we have to deal with can be stated simply: it is that of combining two entirely different media in time.

Each medium has its own multiple parameter sets. The authors' experience suggests that putting

*The authors would like to acknowledge the work of numerous students who have contributed to the work described in this paper through their project work over the years, particularly Alix Smith and Carl Rogers. A Web site related to the work described in this paper can be found at http://www-users.york.ac.uk/~adh2/midas/midas.html

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

together the two media produces combinatorial relationships of such complexity that composers will *need* to develop extensive *algorithmic* control in order to maintain stylistic consistency. This is a central hypothesis of our approach.

2. A CONCEPTUAL MODEL

The work described in this paper is based on a simple generic model of a delivery system for audiovisual compositions which is general enough to encompass traditional performance. It also permits future approaches to the compositional and performance processes for audiovisual works. In this section we present the framework for the model. We then present two examples illustrating its use, and finally we discuss the notion of ‘score’ within the model.

At the core of the framework, shown in figure 1, is the composer’s *script*. This script is a *statement of the composer’s intentions* for the entire system. It may be an implicit entity, consisting essentially of a set of notions and concepts existing within the mind of a composer for the structure of a composition. Alternatively, the script might be entirely explicit, perhaps written in some form of computer language, running on a computer.

In this paper we shall describe a working system which is capable of supporting all of the operational modes of this conceptual model. To do this, it needs to be able to render audiovisual products interactively in real time, although this should not be taken as precluding nonreal time, noninteractive modes of use. The framework comprises a computer system with software capable of running the composer’s script and rendering high-quality sound in real time to an array of loudspeakers, while simultaneously generating graphical output (also in real time) on a projected screen or console. The script permits the detailed synchronisation of sound and image, if required. It can also be used to dynamically define the *degree of freedom* assigned to a performer to determine the subsequent evolution of the audiovisual piece, within a framework laid down in the script by the composer.

For run-time performance reasons, it is likely that the rendering of the sound on the one hand and the graphical output on the other will be handled by separate subsystems. The framework described in this paper therefore includes a distributed processing model capable of supporting the rendering of a synchronised audiovisual composition across a network of distributed processors.

2.1. Two examples of the conceptual model

Let us take two areas of the application of the framework we have just outlined. Each represents a form of computer-mediated algorithmic audiovisual work.

2.1.1. Audiovisual composition

This is a self-contained script which contains the control data for the generation over time of an audiovisual work. The interactive ‘performer’ element in figure 1 is thus not required in this case. The sound

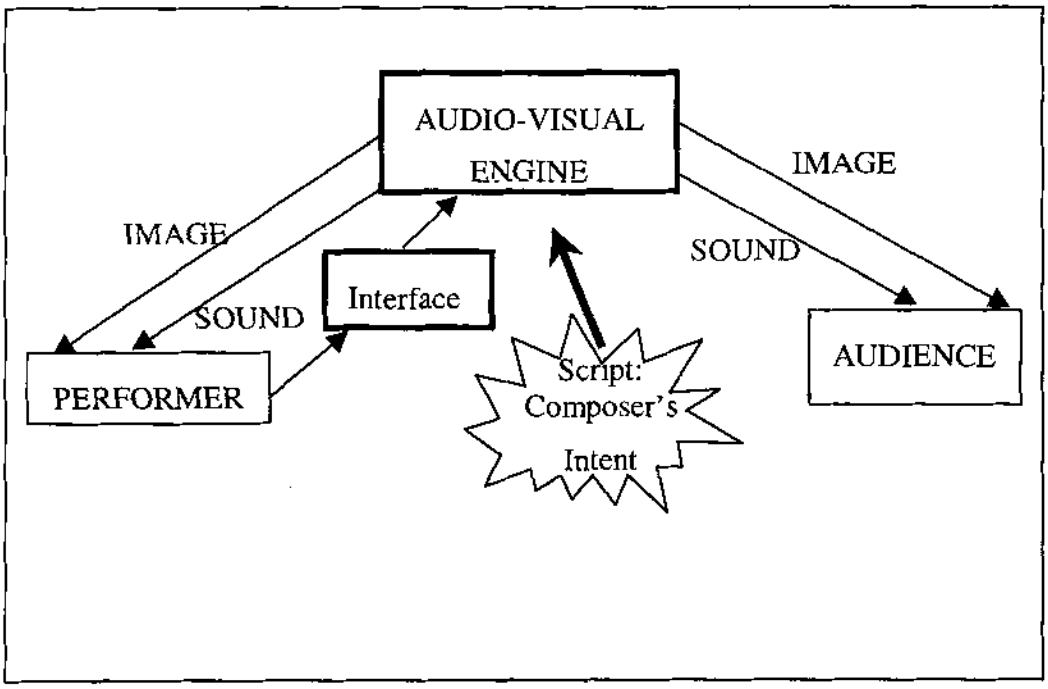


Figure 1. A conceptual model for an audiovisual delivery system.

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

may be electroacoustic in origin, consisting of real-time sound synthesis using well-known techniques (oscillators, filters, vocoder analysis/resynthesis, etc.) and some novel ones (custom treatments of stored sounds). The parameters for the evolving sound processes are generated or delivered by the script from internal data or are extracted from available data files.

The evolving visual elements, which may be abstract or representational according to the intent of the composer, are created from a library of visual generating or processing functions which can be rendered at high speed on dedicated hardware. They include geometric shapes of any size or colour, stochastic rendering functions, and a group of transformation functions permitting intermodulations or interpolations within the available technical resources. Some examples of these are given in figure 2.

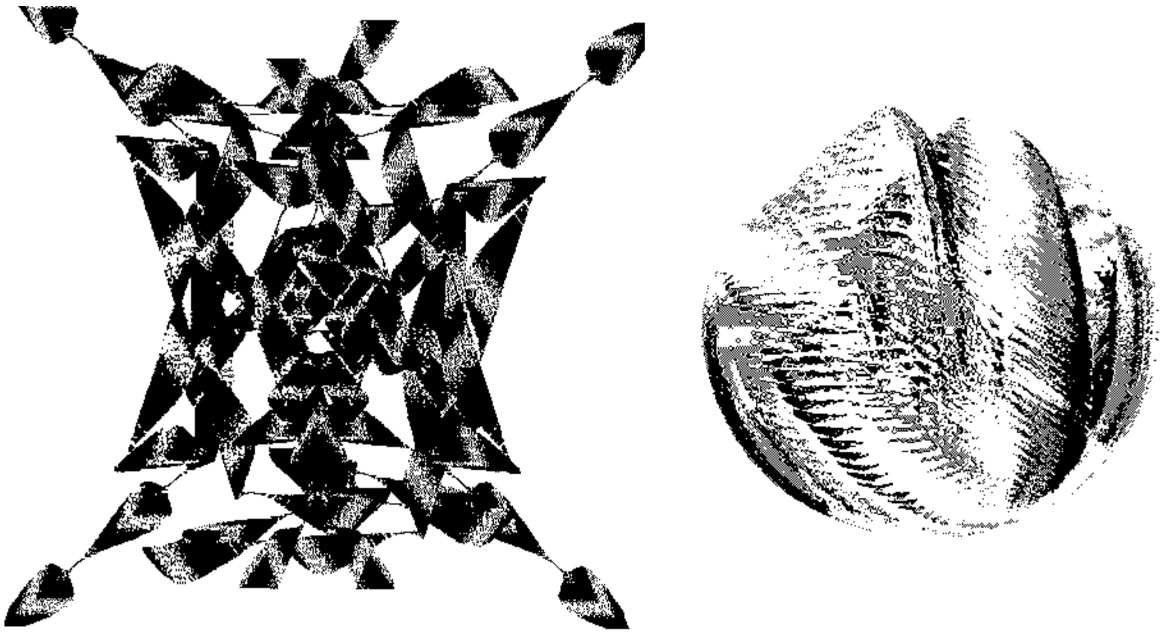
The audiovisual composition may be fully determined in its graphical and sonic evolution – i.e. each performance of the work will be identical and will be of a determined duration. Alternatively, it may run indefinitely through a repertoire of pre-envisioned scenas, while the coupling of image to sound may be tight, loose or continually varying according to controls embedded within the script.

2.1.2. Interactive performance environment

In this model the performer is able to interact with the system to change the sonic or graphical evolution or their coupling, within a framework defined by the composer in the script. The script permits the performer to influence directly the evolution of the graphic development or the evolution of the sound generation and processing repertoire. (The roles of 'audience' and 'performer' might be made deliberately ambivalent in a case such as this.) The script also determines the extent to which the sound and image are affected by changes introduced by the interactive performance element. Note that the actual performer interface may or may not be a standard musical interface (e.g. a MIDI keyboard). We envisage several alternative performance interfaces, a favoured example being a sphere (the 'orb') held between the hands, sensing the pressure of fingers in cardinal directions, transmitting direct or derived control data to a remote receiver. The script determines how the data is mapped to the sound or graphical unit generation.

2.2. The notion of 'score' within the generic model

The concept of 'score' within the model should be examined a little more carefully at this point. There are many manifestations of the role of the score within the model, as we describe below. Sounds and images are produced as the outcome of the composer's intent – either directly as encoded within the script or via a performer's intervention, mediated by the script. These outcomes are perceived both by the audience and by the performer, and may be used for



(a) (b)

Figure 2. (a) Image based on mirrored transformations and replications of a simple geometrical shape (triangle); (b) OpenGL 3D rotational rendering of a photograph of a lion.

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

different purposes within these contexts. For instance, a performer may regard a visual outcome as a score providing 'cues' for his or her subsequent actions.

This kind of usage of a score implies a degree of interactivity on the part of the performer. There could be a process of *passive-score* interaction where the performer essentially *maps* representations contained within the score onto another object, such as an instrument. This is the process of conventional musical performance with orchestral instruments, for example. Alternatively, the model allows a process of *active-score* interaction where the performer directly interacts with the audiovisual output of the piece (the score), influencing its subsequent evolution in a direct or indirect way. This latter case requires the provision of new kinds of *interactive* image rendering systems (for the visual case), such as that described later in this paper.

In the case that the graphics represent a form of performance score for the performer, there may be a 'time line' representing 'now', with graphical elements (generated by a combination of the script and previous performance actions) scrolling across the screen. Alternatively, borrowing concepts from virtual reality, the graphics may flow around and over the performer in an immersive rendering environment, with 'now' representing a point in virtual space. In either case, the graphics indicate through a symbolic mapping the next actions the performer should take.

An alternative conception of the score may be that of a *study-score*, indicating in a visual manner, for instance, sound relationships evolving through time, making certain relationships more directly appreciable by an audience. We certainly do not underestimate the aesthetic value of an evolving realtime score which reflects the sound-generating process of the composition. Therefore, the model encompasses the concept that the 'score' might be a pleasing visual artefact in its own right, and might be perceived as such by an audience.

It is recognised that many different types of representation are possible, not all of them accessible to realtime generation. Even where time-critical analytical processes are required, we should note that they may take place 'out of time' and stored for display during a later run of the composition.

Somewhat more remote from normal musical practice is the notion of the sonic output being a 'score' for some other, parallel artistic activity. One relatively easily comprehensible notion is where the sound acts as a 'score' for a dancer, where sets of sonic symbols are indicators for the type of dance required by the composer/choreographer.

We recognise that the *generator* of the audiovisual performance in the model we have adopted, the *script*, is in some sense a score. However, the script is not part of the graphical output of the system, and what we are considering here is the use of the system's display as some form of alternative score. This score may or may not be part of the generating loop, providing interactive visual feedback for the performer. Any situation which demands the creation of a parallel visual score-representation to a realtime sonic performance requires there to be an understood *consistency* of the relationship between the two media. Such a consistent relationship may be an asset, or it may be a hindrance, for multimedia artists on the whole, who may wish to eschew consistency in favour of surreal discontinuity.

In this sense it must be seen that the notion of a score is a *special case* of the relationship between the two media. The notion of a traditional score fits perfectly into the model shown in figure 1. In this case the composer's script says, '*Run the script just once; produce a fixed graphical output in common music notation; then let the performer's gestures drive the instrument directly without any further intervention by the composer*.'

3. THE ALGORITHMIC IMPERATIVE

We are of the belief that *all* composition is algorithmic in origin, whether the composer is aware of it or not. In other words, the composer may consciously choose to adopt algorithmic methods or he may not – but in the latter case the algorithmic source remains at a subconscious level. Even the random selection of elements and processes is an algorithm. In the extreme case, we might detect the 'null algorithm', which is a refusal to entertain any algorithmic process; in this case the choice of any content is left entirely to chance. However, a decision to avoid any decision is still a decision, and lies at one end of the spectrum of algorithmic prescription. We are therefore not concerned with justifying an algorithmic approach to the problem of musical and image organisation. We take this as a necessity, leading to the requirement for a comprehensive organisational network for algorithmic multimedia composition in general.

It follows that the complexities of parametrical relationships both within each contributing medium, and also the additional interrelationships of the two media, require management by an algorithmic language. The requirements for a language which will support the conceptual model described in this paper are that it permits simultaneous and coordinated generation and processing of electroacoustic sound together with generation, processing and rendering of images interactively in real time. In addition, processed and/or reproduced recorded sounds and images will be included in the repertory of available resources. The algorithmic features of the language must include precise time coordination both within and between the media, as well as a full set of mathematical routines for creating abstract time-evolving functions which can then be mapped onto appropriate sonic and visual domains.

One of the consequences of this decision is that great flexibility in the treatment of algorithmic processes is allowed. Certain parts of the organisation of a work may be rigid in treatment, while others may be permitted great freedom of selection by the composer (or performer), or determined from random or quasi-random principles. The relationship between those parts of the organisation that are flexible and those that are rigid does not need to be static; it can itself evolve over time. Thus we envisage great expressive capability from the users of such a language.

4. A PRACTICAL APPROACH TO THE IMPLEMENTATION OF A MULTIMEDIA COMPOSING ENVIRONMENT

The algorithmic composition language Tabula Vigilans (Orton and Kirk 1992, Orton 1993a–1998b) and the multiprocessor signal processing environment MIDAS (Kirk, Whittington, Hunt and Orton 1995, Kirk and Hunt 1996) have evolved in parallel since 1992. We believe they embody the necessary principles such that their merging and development will support the comprehensive requirements of the algorithmic audiovisual (multimedia) system we have outlined in the earlier part of this paper.

4.1. Tabula Vigilans Audio Interactive (TVAI)

Tabula Vigilans (Orton 1994a) is a composer's scripting language which, up until recently, has used MIDI to produce sound output for a high-level composing utility. Tabula Vigilans Audio Interactive – TVAI – is an extension of the original Tabula Vigilans language to incorporate sound synthesis and processing capability and also image generation and processing. These extensions employ the MIDAS platform which is described in more detail in the next section. TVAI is currently under development and at the time of writing (December 1998) the following sound-synthesis-generating and image-generating rules are available:

oscil oscillator UGP, with a choice of waveform
parameters: amplitude, frequency
waveform: sine, cosine, triangle, square, pulse, sampled from soundfile

grain specify grain type for granular synthesis
parameters: amplitude, grain size (e.g. 15 ms), envelope type, waveform type

stream create a granular synthesis stream with specified parameters
parameters: amplitude, density, degree of overlap, centre frequency, frequency bandwidth, regularity

rectangle create a rectangle with specified parameters
parameters: window location (x and y coordinates), size, colour

Within the Tabula Vigilans terminology, each of these keywords is a *primitive rule* which occupies one line of the composer's script. The **rectangle** rule will shortly be replaced by the more generic **polygon** rule, with an additional parameter of *number of sides*.

There is not space here to list all of the features of TVAI which make it flexible for composerly control. The main algorithmic features may be studied in the Tabula Vigilans User Manual, which is in the public domain. Suffice it to say that many table-based operations over time may be employed, some of which are akin to high-level event lists, others permitting efficient recursive or constrained algorithmic iteration.

To give a simple illustration of the use of a script to generate both sound and image simultaneously, figure 3 shows a short but complete script to produce evolving granular streams and images. A variant of this script may be used to generate the kinds of image shown in figure 4. However, since random decisions are used at many points in the script, no two performances will generate exactly the same image. The script simultaneously creates a granular audio stream with continuously varying densities, centre frequency, bandwidth and regularity. In this script, the parameters for image and signal generation are separate; however, the TVAI language makes it equally possible to have the two media generated from the same algorithm and mapped onto desired control ranges for image and sound generation.

4.2. The MIDAS audiovisual synthesis platform

This section describes work which extends the unit generator approach for the synthesis of sound to include synthesis of image. This allows image and sound to be generated in one consistent environment, where an audiovisual composer or performer can carefully control the interrelationship of these two elements, including synchronisation. This environment can be controlled and coordinated from TVAI.

```
//////////////////////////////////////////////////////////////
//     Tabula Vigilans Audio Interactive demonstration script
//     employing granular synthesis routines and creation of
//     images using rectangles.
//
//     © Richard Orton York 1998
//////////////////////////////////////////////////////////////
start()
{
       sr           44100                   // set sampling rate
       grain        1.0, 0.015, 3, 0        // maxamp, grainsize 15mS
                                            // cosine waveform
       openwindow   1280, 1000              // open window 1280 x 1000
       call init_rect_params()
       while (t < 360) {                    // for 360 secs
              t time                   // set clock time
              call set_stream_params()
              stream 1.0, freq, dens, bw, reg // generate grain stream
              rectangle x, y, xsiz, ysiz, col // generate rectangle
              call next_rectangle_params()
       }
}

init_rect_params()                // set initial rectangle position
{
       xsiz = int(random(10, 100))
       ysiz = int(random(10, 100))
       x = int(random(xsiz, 1280 - xsiz))
       y = int(random(ysiz, 1000 - ysiz))
       col = 0
}

set_stream_params()               // adjust granular stream
{
       freq   lin 10, 0, 300, 800 // set centre freq to
                                       // move from 300 - 800 Hz
                                       // and back over 10secs
       bw     lin 8, 0, 0.1, 2.0  // set bandwidth to move
                                       // from 0.1 octave to 2.0
                                       // octaves over 8secs
       dens   lin    15, 0, 1, 100     // Density cycle: from 1
                                       // event per sec to 100
                                       // events per sec each 15s
       reg    lin 12, 1                // regularity cycle:
                                       // from maximum
                                       // irregularity to max
                                       // regularity over 12s
}

next_rectangle_params()                // adjust rectangle
{
              xsiz += int(random(-10, 10))
              xsiz lim 10, 50
              ysiz += int(random(-10, 10))
              ysiz lim 10, 50
              x += int(random(-20, 20))
              x lim xsiz, 1280 - xsiz
              y += int(random(-20, 20))
              y lim ysiz, 1000 - ysiz
```

Figure 3. Tabula Vigilans Audio Interactive script example.

The TVAI/MIDAS platform directly supports the generic model described in section 2 above.

The unit generator paradigm is a tried and trusted means of synthesising sound and has formed a basis for the realisation of electroacoustic composition and performance for many years. A number of computer music environments (e.g. the Csound system) are now widely available which use this paradigm. These environments provide a scheduling and data interchange kernel which manages the interaction between signal processing objects (the unit generators) whose configuration can be defined by the musician. This gives very flexible control over the music synthesis algorithms.

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

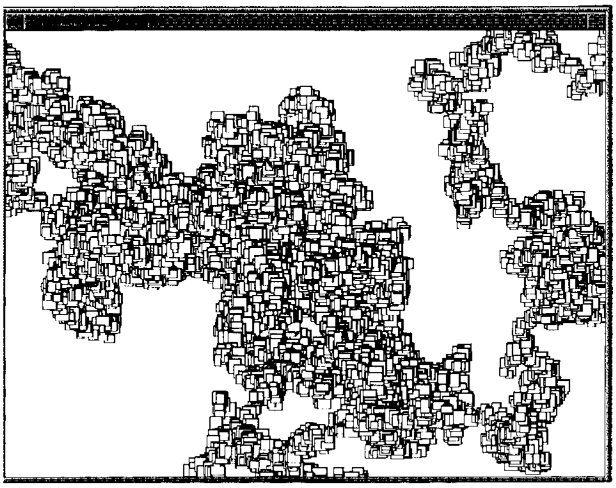


Figure 4. An example screen-shot produced by the TVAI script shown in figure 3.

As stated above, our interest lies in exploring the use of the unit generator approach to include *graphical image* as a medium for composition and performance, so that we may construct audiovisual instruments built from integrated networks of sound and image unit generators.

In the work described in this paper, the University of York's MIDAS system (Kirk and Hunt 1996) is used as a platform to run audiovisual unit generators, which are known as unit generator processes (UGPs). MIDAS is designed to provide a scalable architecture oriented towards the support of the signal processing algorithms used in the synthesis of sound, within a regime of hard realtime deadlines. The ear lacks the integrating nature of the eye, and is therefore more critical of discontinuities in processing performance. An architecture designed for the synthesis of sound is therefore seen as a useful starting point for applications where synchronisation of sound with image is a significant factor, especially when the two can be integrated into one coherent framework within the architecture.

4.2.1. Architectural description of the MIDAS system

The core of the MIDAS system is a form of Local Area Network, known as MILAN (Musical Instrument Local Area Network). It is designed to support dynamically defined synchronous signal processing algorithms, whose elements (UGPs) can be distributed across the heterogeneous processing nodes forming the network. MILAN provides a set of protocols which allows uniquely identified UGPs to be instantiated on a specified node, within a scheduling and synchronisation primitive known as a *virtual processor*. Other protocol elements allow unit generators to be connected together (and hence interchange data) within a virtual processor, and also allow virtual processors to be connected together over *virtual channels*. These virtual channels may exist within one physical processor on the network, or alternatively, may be mapped onto the physical network segments within MILAN. Data may then be exchanged between virtual processors over the virtual channels using data transmission protocol elements. These concepts are illustrated in figure 5, where MILAN protocol messages have been used to construct a signal-processing algorithm consisting of UGPs U_1–U_6. The UGPs have been located within two virtual processors VP_1 and VP_2 placed on two physical processors P_1 and P_2 communicating across MILAN.

More than one virtual processor can be located within a physical processor. Other protocol elements are used to dismantle a network of UGPs when it is no longer required. Each physical processor runs a kernel process which deals with any messages it receives. The kernel then runs down a linked-list of virtual processors allocated to it, to see whether any of them are ready to run, in accordance with the principle of data-flow synchronisation described below. The operation of the kernel ensures that the networks of UGPs and virtual processors can be built and dismantled *dynamically* at run-time.

Data flow synchronisation is used to prevent UGPs within one virtual processor from running ahead of others. A virtual processor is suspended until data is present on all of its inputs. A datum is then consumed from each input, which is then fed into the UGP network within the virtual processor. The UGPs within the virtual processor are run in a fixed order, which ensures that the data is passed down the network of UGPs in an appropriately ordered manner. Each input in a virtual processor is provided with a FIFO buffer. This retains data, which arrives at a virtual processor ahead of other time-related data which will arrive later, having been processed via a more extended chain of physical processors. The buffers are bounded by allocating only as much UGP load to a processor as it can handle in one unit sample 'tick'. Data therefore flows through the network in a systolic data flow, where UGPs in an algorithmic network are operated top-to-bottom, in lock-step synchronisation. For high-quality audio signals (comparable to CD or DAT), the interval between successive samples is of the order of 20 μs, for a 16 or 32 bit data word exchanged between a single pair of UGPs. Since many such paths can exist within a virtual channel, and since many virtual channels can exist on each physical segment of MILAN, it is clear that network segments are required to handle high data rates.

It is important to realise that a signal processing algorithm is defined in terms of a synchronised network of UGPs (i.e. not in terms of native code for a

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

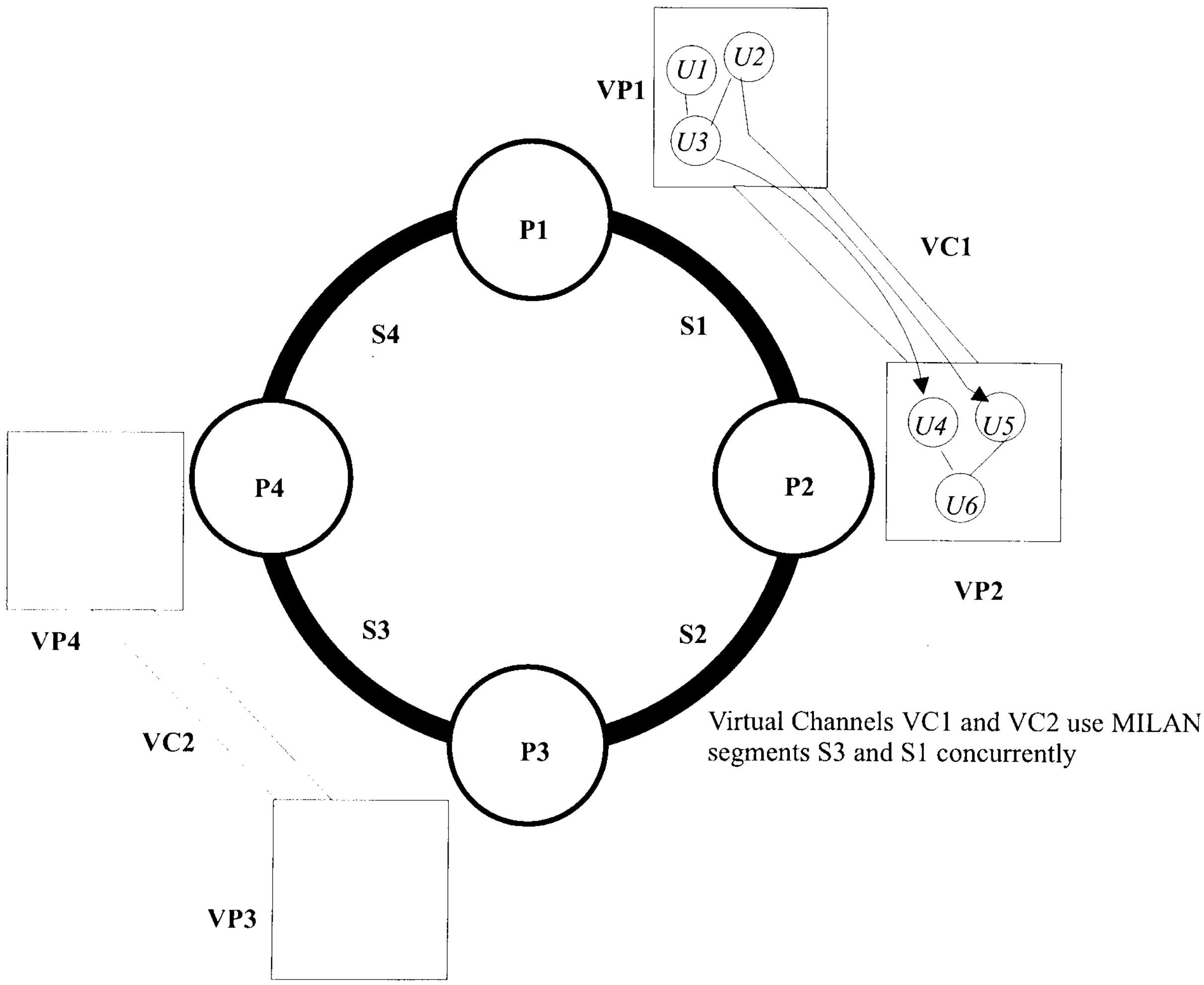


Figure 5. The MIDAS–MILAN System.

specific processor), and that these networks are constructed and controlled by means of the protocols. This means that as long as a physical processor can interpret and react properly to the protocols, and as long as it can support the UGPs it is asked to provide, the processor can be of any type. In particular, heterogeneous processors can be integrated into the network. New processors can be incorporated as they are introduced, simply by writing an appropriate protocol interface and UGP library. Algorithms (defined in terms of the protocols) will continue to run without modification (just with higher performance). This avoids the problem of closed hardware design and system obsolescence, which has bedevilled high-end music systems in the past. In principle, given the lightweight nature of the protocols, processing nodes could be specialised hardware devices (i.e. not conventional processors), optimised for specific audio and image processing tasks.

4.2.2. Implementation of graphical unit generator processes

UGP libraries have been produced which contain many of the signal processing primitives found in music synthesis systems such as oscillators, filters, envelope generators, mixers, gates and general algorithmic components such as arithmetic operators and predicate UGPs. To these, we have added graphical primitives, including WIMP facilities, so that integrated networks of signal processing and graphics algorithms can be produced within one coherent environment. Since the objective of the MIDAS system is to support realtime *performance* of electroacoustic music, adding a graphical dimension to a UGP network opens up the possibility of *graphical as well as sonic performance*.

The graphical UGPs are based on a portable graphics library which has been used for graphical applications running on a number of machines, including PCs and Silicon Graphics workstations. Synchronised MIDAS audiovisual applications may be distributed across multiple processors using sockets. The library, based on Open GL, provides common graphical objects (such as lines, rectangles, mouse click buttons, and mouse coordinate functions) and more sophisticated 2D and 3D rendering tools such as texture mapping, camera animation,

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

lighting effects and image buffering techniques (stencil, accumulation, depth, etc.). Figure 2(b) is an example of this rendering facility in MIDAS, based on a single static image of a lion club undergoing various OpenGL-based rotational transformations (Ash 1999).

Since the UGPs are operating continuously within the MIDAS environment, the user may interact with the images produced. Some of these dynamic properties include the position and size of the graphical objects. It is in principle possible to grab any of the images with the mouse, and to move it to a different position within the screen canvas. This of course would not always be desirable in a control panel application, where we probably want sliders, etc., to be firmly fixed in position! However, the dynamic operation of the MIDAS system means that we may be able to exploit these aspects in other kinds of visual output.

In particular, we have explored the possibility of synthesising dynamically *evolving* images constructed from step-and-replicate operations with simple geometrical objects such as rectangles, circles and triangles, of varying size, position, colour and line style. The graphical UGPs are provided with inputs that will accept numeric parameters and which control these graphical attributes. If the UGPs are placed in an algorithmic UGP network which generates time-varying parameter flows in accordance with the wishes of a visual composer or performer, then the evolution of the images can be controlled in a composerly way to create a visual output.

Figure 2(a) shows an image generated by MIDAS which adopts this kind of higher-order approach to image synthesis. In this image we have used a meta-image generator UGP, based on the principles of the kaleidoscope. It drives the 'position', 'size' and 'colour' inputs of a simple triangle generator. Images of triangles are drawn either side of two lines of symmetry. However, the symmetry of the image can be *skewed* by inputs to the kaleidoscope UGP which add offsets to the symmetrical drawing processes. These offsets are placed under algorithmic control in the UGP network. It will be noticed in figure 2(a) that although the image clearly has a high degree of symmetry, representing the principle of the kaleidoscope, there are also subtle variations in symmetry, representing the skewing effects. These variations could be made more or less gross, depending on the wishes of the composer or performer. The image is continuously evolving in real time, under the control of a performer. The image shown in figure 2(a) is a snapshot taken about one minute into the performance on a Silicon Graphics Indy running MIDAS, and was continuing to evolve at this time. Many different kinds of interactive image of this nature can be produced. The kaleidoscope is only one of a large number of algorithmic sound/image generators which could be placed under the control of TVAI.

5. THE ART OF AUDIOVISUAL PERFORMANCE

It should be emphasised that much of the dynamic visual interest of the images is lost by the need to present static 'snapshots' of the graphical output on the printed page. A considerable part of the aesthetic interest lies in the evolution of the images. This dynamic property can be enhanced by a manipulation of the *history* of the evolved image. For instance, images of primitives could be replaced by modified instances (e.g. changed colour), or even removed, as a function of the number of iterations in elapsed time, to create a visual analogue of reverberation.

Since the MIDAS system allows *any* unit generator to be instantiated on another processor in the network, it becomes possible for one performer to open a window and construct interactive images of the form shown in figure 2 in another performer's image space on his or her audiovisual workstation. The second performer might then be invited to interact directly with this remotely instantiated image in a way which redefines the overall audiovisual output. This might be the beginning of an audiovisual orchestra.

If such interactive image manipulation is to form part of performance, some basic questions regarding the nature of the visual product need to be explored and resolved. The *viewer* might perceive an image as a holistic phenomenon, taking it in with a single sweep of the image field. By contrast, sound is largely a serial phenomenon; a dynamic characteristic which perhaps lends itself to expression within performance.

With respect to the visual performance there are two major issues which should be considered. Firstly, our experience so far has shown that the *artist* using such a dynamic image creation system is 'engaged' by the dynamics of the process – i.e. the artist enjoys the feeling of control. Does this enjoyment necessarily pass on to a third-party observer who is not in the control loop? Secondly, we need to ask whether the dynamics of the performance have perceptual merit in their own right, or whether they only make sense in the context of the composite image, and if so, what is the relationship between that context and the performance dynamic?

Whatever the conclusions regarding these matters, we need to consider the nature of the *algorithm* with which the performer is interacting. It is a trivial matter to construct algorithms which have nonlinear, and even chaotic behaviours within the unit generator structure, and therefore within the visual domain. The interactive facility provided within MIDAS

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

allows a visual composer to explore a chaotic parameter region associated with an image. Often, a region of visual coherence will be discovered by this interactive process. The parameter set corresponding to this coherent scene can then be used as a basis for further development. The phoenix is roused and then tamed.

The computer, in this paradigm, is purely a tool to facilitate the creative process in exactly the same way as artists have been using tools since time immemorial. Careful consideration of the interface design between the artist and the computer will result in a tool with a multifaceted character. Exploration of the juxtaposition of algorithmic structure and interface design will enable artists to develop an aesthetic language for audiovisual composition.

Figures 6(a) and (b) were created using the MIDAS UGP 'WhitneyPaint' (Merrison 1999). The genesis and concept of WhitneyPaint used the techniques described above. It was written both to take advantage of the MIDAS system (allowing exploration of cross parametric mappings) and to explore the audiovisual theories and techniques of John Whitney (Whitney 1980), in particular his concept of differential dynamics using polar plotting algorithms. These images are based on the coordinate transformation

$$x = r \sin \theta, \qquad y = r \tan \theta.$$

Once a coherent base image had been discovered interactively, it was then mutated in an improvisational sense, by applying constrained transformations to angular rotations of θ and line width, etc. Although not used in the generation of this image, there is no reason why scored transformational processes could not be used, employing Tabula Vigilans.

6. IMPLICATIONS FOR THE DELIVERY OF AUDIOVISUAL CONTENT

In this paper we have described a scripting language (TVAI) driving an audiovisual rendering engine (MIDAS) capable of realtime interactive synthesis of image and sound. The scripting language is capable of expressing the composer's intent within the audiovisual performance.

This arrangement may be interpreted in a different way within the context of contemporary multimedia systems, where audiovisual 'content' is distributed via the World Wide Web. The issue at stake in such systems is the problem of delivering large amounts of (audiovisual) information through a limited bandwidth channel. Even though the information capacity of such networks is predicted to grow very substantially in the near future, so is the amount of traffic which will compete for the network resource. The problem of delivery of high-density information via the network is exacerbated when the content has an *interactive* component, which implies a realtime rendering activity. It is no longer adequate to deliver fixed-form (prerendered) compressed information across the net.

A more appropriate approach would be to deliver a highly abstracted description across the net, driving a local interactive rendering engine. The abstract description specifies the required operation at a higher level than that used by the rendering engine, and its transmission across the net would therefore

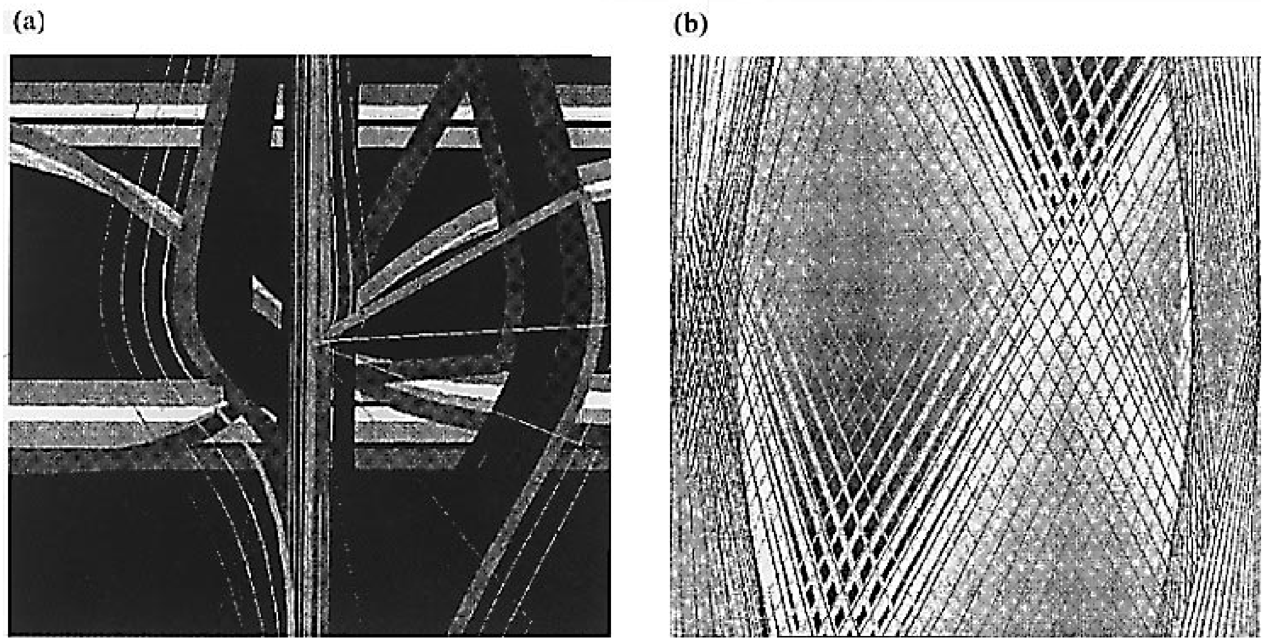


Figure 6. Two examples of images rendered with MIDAS UGPs based on the transformation: $x = r \sin \theta$, $y = r \tan \theta$.

https://doi.org/10.1017/S1355771898003045 Published online by Cambridge University Press

typically require substantially less network bandwidth. Within the context of the work described in this paper, TVAI scripts provide the high-level abstract description of an audiovisual piece, and a derivative of MIDAS could provide the local rendering engine.

We envisage the local computer host (or 'rendering black box') being connected to a World Wide Web source of multimedia performance scripts which will provide a virtually limitless source of new audiovisual compositions, which may be interactive where desired. (Even interactive compositions will be provided with a noninteractive mode so that a performance still results when there is minimal or no user input.) The domestic user will download a new composition from the Web and begin a performance. Special large flat display screens will be available, or the television monitor may be used. The performance is mediated from the host computer, or the rendering 'black box' which contains the sound and graphics rendering hardware in response to the algorithmic controls dictated by the embedded script. Advanced users may generate their own multimedia performance scripts, but more likely is the modification of existing scripts to the taste of the user, according to templates which the originating composer has provided. A special graphical interface will be provided to facilitate such modifications without the user needing to be a proficient programmer.

7. CONCLUSIONS AND FURTHER WORK

The merging of an algorithmic composition language, Tabula Vigilans, with the MIDAS multiprocessor sound synthesis and graphical rendering engine has enabled some first attempts at realtime audiovisual composition. Now that we have a basic set of interactive audiovisual tools, we need to work with artists/animators to develop an aesthetic and praxis for this new performance and composition medium.

We feel that we are at the beginning of a new venture, where the technical issues of image generation and coordination with sound, which although challenging, may be relatively minor. The questions regarding the praxis, aesthetics and semantics of performance with image and sound may, ultimately, provide the greater challenge.

REFERENCES

Ash, K. 1999. *MSc Thesis in Music Technology*. University of York, September 1999.

Kirk, R., Abbotson, M., Abbotson, R., Hunt, A., and Cleaton, A. 1994. Computer music in the service of music therapy: the MIDIGRID and MIDICREATOR systems. *Medical Engineering and Physics* **16**: 253.

Kirk, R., and Hunt, A. 1996. The MIDAS–MILAN system. *Journal of the Audio Engineering Society* **44**(3):119.

Kirk, R., and Orton, R. 1994. Evolution of timbres through the use of *Tabula Vigilans* on the MIDAS system. In S. Emmerson (ed.) *Contemporary Music Review* **10**(2): 201–9. Switzerland: Harwood Academic Publishers.

Kirk, R., Whittington, P., Hunt, A., and Orton, R. 1995. Graphical control of unit generator processes on the MIDAS system: a digital VCS-3 demonstrator. *Proc. of the ICMC*, p. 499. Banff, Alberta.

Merrison, B. 1999. *MSc Thesis in Music Technology*. University of York, September 1999.

Orton, R. 1993a. Musical applications of the tabular manipulations in *Tabula Vigilans*. *Chroma*, *Journal of the Australian Computer Music Association*.

Orton, R. 1993*b*. *Timeshadows*. *Tabula Vigilans* composition first performed at the Australian International Symposium on Computer Music, Sydney 1993.

Orton, R. 1994a. *Tabula Vigilans User Manual*, Version 1.0. Composers' Desktop Project.

Orton, R. 1994b. *Ceci n'est pas un Orchestre*. *Tabula Vigilans* composition first performed at the 'Sound Encounters' Festival of New Music, Cleveland, OH, 1994.

Orton, R. 1995. *Gloria* for Keyboard and Interactive *Tabula Vigilans*. First performance, York 1995.

Orton, R. 1996a. Design strategies for algorithmic composition. *Contemporary Music Review* **15**(3–4): 39–48. Amsterdam: Harwood Academic Publishers.

Orton, R. 1996b. *Stellations* for String Orchestra. Commissioned by the European Community Chamber Orchestra. First performance, 1996 (score composed with the aid of Tabula Vigilans).

Orton, R. 1998a. Choice and chance in algorithmic composition. Paper delivered at the *Workshop on Compositional Use of Computers*, University of Limerick.

Orton, R. 1998b. *Tabula Vigilans User Manual*, Version 1.2. Composers' Desktop Project.

Orton, R., and Kirk, P. R. 1992. Tabula Vigilans. *Proc. of the Int. Computer Music Conf.* 1992, pp. 243–6. San Francisco: International Computer Music Association.

Whitney, J. 1980. *Digital Harmony*, Chap. 5. McGraw-Hill.