

# **Bats, Max, Boids and Music**

## **Ultrasonics, Artificial Birds, and Max Programming in a non-tactile Musical Environment**

**Steven Campbell**

*James Cook University*  
*[steven.campbell@jcu.edu.au](mailto:steven.campbell@jcu.edu.au)*

**Abstract** Bats, Max, Boids and Music, are each elements that contribute to M-SEA, a music technology system in development since 1999. M-SEA, standing for Multi-Sensor Environment for Audio, is a non-tactile system that uses ultrasonic sensing in a 100sqm (or smaller) area to detect locations and activity levels of people moving within the space. Sensor data is converted to digital data by a proprietary M-SEA hardware interface and relayed to the hub of the system, an Apple computer running M-SEA's proprietary software. The software uses the digital data to manipulate musical materials and parameters, the resulting music played back in real-time to those moving within the space. This paper presents the 'Digital Voyage' undertaken to develop and explore each of the system elements listed above, and provides an overview of the M-SEA hardware and software. Brief descriptions of music programs created for the system are incorporated.

### **1. Introduction**

The Multi-Sensor Environment for Audio (M-SEA) began its development in 1998, following a visit to Australia by UK based composer/music technologist Rolf Gehlhaar. Gehlhaar brought with him his SOUND=SPACE, an ultrasonic system that relayed measurements from ultrasonic sensors to an Atari computer (Gehlhaar, 1991). The data was then used to control musical parameters, playing music back to people moving in front of the sensors. Numerous applications of the Gehlhaar system were demonstrated and discussed by the composer, these including new music composition, workshops with disabled children (Gehlhaar, 1998), performances in dance, and gallery and museum installations. Following the visit, there was considerable enthusiasm in the community, particularly the disabled/special education community, to attain such a system. However, at a cost of c.\$35,000, the purchase of the Gehlhaar system was prohibitive.

Given this prohibitive cost, a number of commercially available ultrasonic systems were investigated by the author, with a view to assembling a similar system and utilising a knowledge of Macintosh music programming software to provide software for the system. A grant was received from James Cook University in 1999 to purchase commercially available hardware, followed in 2000 by another from the Apple University Development Fund to continue the work. Numerous problems became evident with the commercial sensing hardware tested, these problems subsequently solved in collaboration with Dr Graham Woods (of James Cook University's Electrical Engineering Department) with the development of a system utilising a combination of commercial and proprietary hardware. The first working version of M-SEA was successfully tested in 2002.

An overview of the system hardware and software is related in this paper. The paper provides an analogy of the system's ultrasonic basis with that of bats, followed by a background to the Max programming environment in which the M-SEA software has been developed. An overview is provided of a virtual simulation system used to represent people moving within the sensing space. A number of music compositions, in the form of software files, have also been written to demonstrate M-SEA's range of musical capabilities. These are incorporated as default files in the M-SEA software and are discussed individually in the final part of this document.

### 1.1 System components

The M-SEA system is comprised of four main parts, as shown in Figure 1:

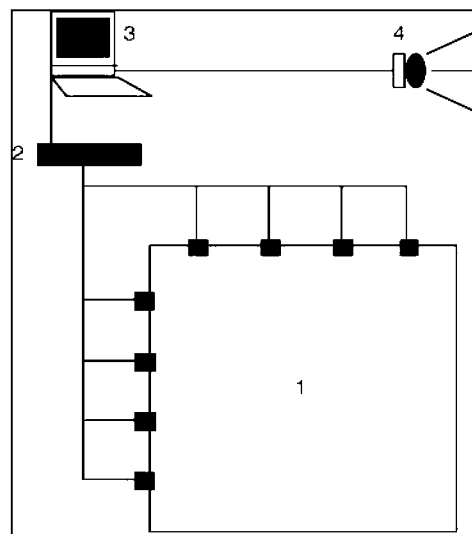


Figure 1

1. eight ultrasonic sensors in a 100sqm (or less) space;
2. a hardware interface that controls sensor timing and converts sensor data to digital data;
3. a Macintosh with proprietary software that a) utilises the data from the interface to create and/or control audio and visual material, and b) provides a simulation of the sensing space;
4. 3<sup>rd</sup> party software and hardware for audio and visual output.

Components 1, 2 and 3 are discussed below. Component 4 is variable, comprised of commercially available music software and hardware components. The current components in this part of the system include a software sampler, various audio plug-ins for audio effects, a Firewire audio interface for digital to analogue audio conversion, and a small PA system for audio output.

Working entirely from the measurement of distances between the sensors and people or objects within the space, M-SEA is a non-tactile system. Unlike in many interactive music systems, M-SEA users are unencumbered with any devices that need to be attached to the body, and therefore users do not have to interact with any further physical components.

## 2. Bats and Ultrasonics

Bats use ultrasonic sensing, or echolocation, for both navigation and the location of prey. In a basic analogy for the MSEA ultrasonic system, eight bats can be imagined placed along two adjacent walls of a room. They are instructed to send out their ultrasonic sounds, wait for the returning reflection or echo of their sounds, and then report the time that it took for the sounds to return to them.

Given that the room is 100sqm and that the speed of sound is fairly constant (c. 330 meters per second), it will take 64 milliseconds for the sounds to travel 10 meters, hit an opposite wall, and return. Reports from each bat will then be “64”, as shown in Figure 2.

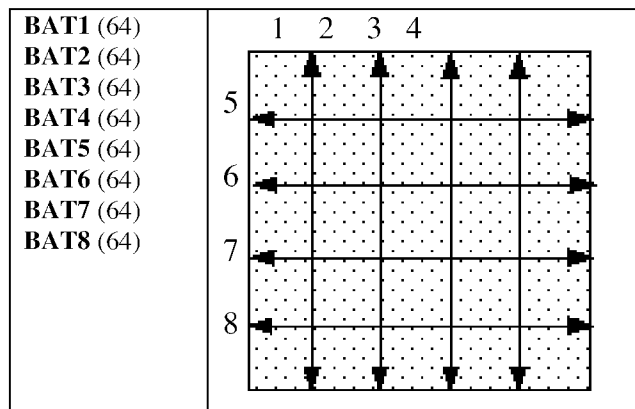


Figure 2

If objects are then placed in the space at varying distances from the bats, varying responses will be obtained, such as those in Figure 3.

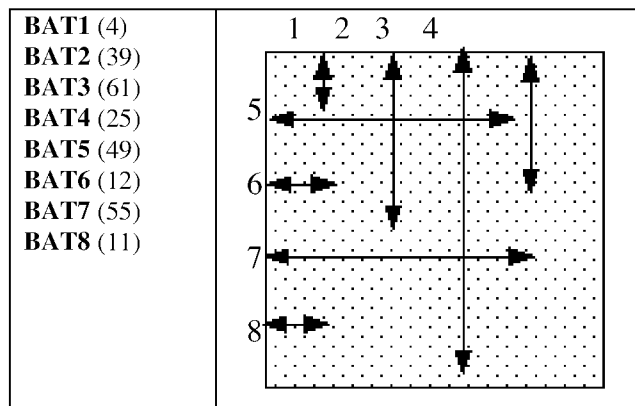


Figure 3

The M-SEA system uses electronic versions of the bats, ultrasonic sensors from Polaroid (Polaroid OEM Components Group, 1999), to perform this process. However these sensors,

unlike their natural counterparts, cannot individualise their own emissions. In the wild, a bat can send various types of sounds, ie constant and modulated frequencies, and can also control the harmonic content of its sounds. Such controls enable a bat to avoid objects in different environments such as in vegetation or in the open air (Suga, 1990). The sounds used by individual bats also differ from one bat to another, much as humans have distinct voices, enabling bats to navigate without interference from other bat sounds.

Electronic sensors are considerably more limited, using a single frequency and timbre. With this limitation the send and return of each sensor's emission must be completed before another sensor can begin an emission, in order to avoid the detection of one sensor's reflected sound by any other sensors.

The current hardware version of the M-SEA system relies on the maximum time for each of the eight sensors to send and receive (64ms) over a 10m distance. At this rate it takes 512ms for the system to receive reports from all eight sensors, ie. one pass of the room is completed every 512ms.

## **2.1 The M-SEA Sensor Interface**

Each M-SEA sensor detects the time taken for an ultrasonic emission to return to the sensor and relays this to the M-SEA hardware interface as a voltage. The interface controls the conversion of voltages to digital data and also the timing of the sensor emissions. To output, the interface maps voltages to a decimal and relays the data to a computer as a MIDI (Musical Instrument Digital Interface) numeric between 0 and 64. Currently the system uses MIDI System Exclusive as the format for the data. The first version of the hardware uses proprietary circuits to control the sensor timing and incorporates an I-Cube System from Infusion Systems (Garncarz et al, 1998) to convert voltages to MIDI data. The interface incorporates MIDI In and Out sockets for sending and receiving MIDI data.

The second version of the hardware is wholly proprietary and currently in development to use a single microchip to control timing and convert incoming data to MIDI. This version improves the system efficiency by avoiding the more cumbersome System Exclusive MIDI format, instead using simple MIDI note-on messages. Additionally the timing of the system is improved by allowing each sensor to send an emission immediately upon the previous sensor having received a returned emission. This is in contrast to the current version where each sensor must wait the maximum time (64ms) of the previous sensor before it can send. Theoretically, if objects were placed immediately in front of each sensor the overall time for the system to receive data from all eight sensors could be as low as 19ms, compared to the current version's consistent 512ms.

## **3. Max**

MIDI data from the M-SEA Interface is relayed to a Macintosh via a separate USB MIDI Interface, the computer utilising proprietary M-SEA software that has been developed within Max. Max is a graphic programming environment for the Macintosh, based in C, that was commercially released for programming MIDI applications in 1990 (Cycling 74, 2003). Throughout the 1990s Max developed considerably, and in 1998 MSP was incorporated into the environment to allow direct work with audio and sound synthesis. In 2002 Jitter was incorporated, extending Max into the realm of video. The environment was ported to Mac

OSX in April, 2003, allowing the continuation of the M-SEA software development in the current Macintosh OS.

M-SEA, as a MIDI based environment, utilises Max, MSP, and Jitter for the development of a range of music and multimedia compositions that interpret data from the ultrasonic hardware in various ways, as discussed later in relation to the compositions themselves. Max is utilised for the management of MIDI data from the sensors, and to control music parameters for output (notes, rhythms, volume and so forth). MSP is used for controlling sound synthesis, audio effects, 3<sup>rd</sup> party audio plug-ins, individual sound samples and loops. Jitter is used to convert MIDI data from the sensors into visual effects, the output to a data projector and thence back into the sensing space, or onto screens at the perimeter of the space.

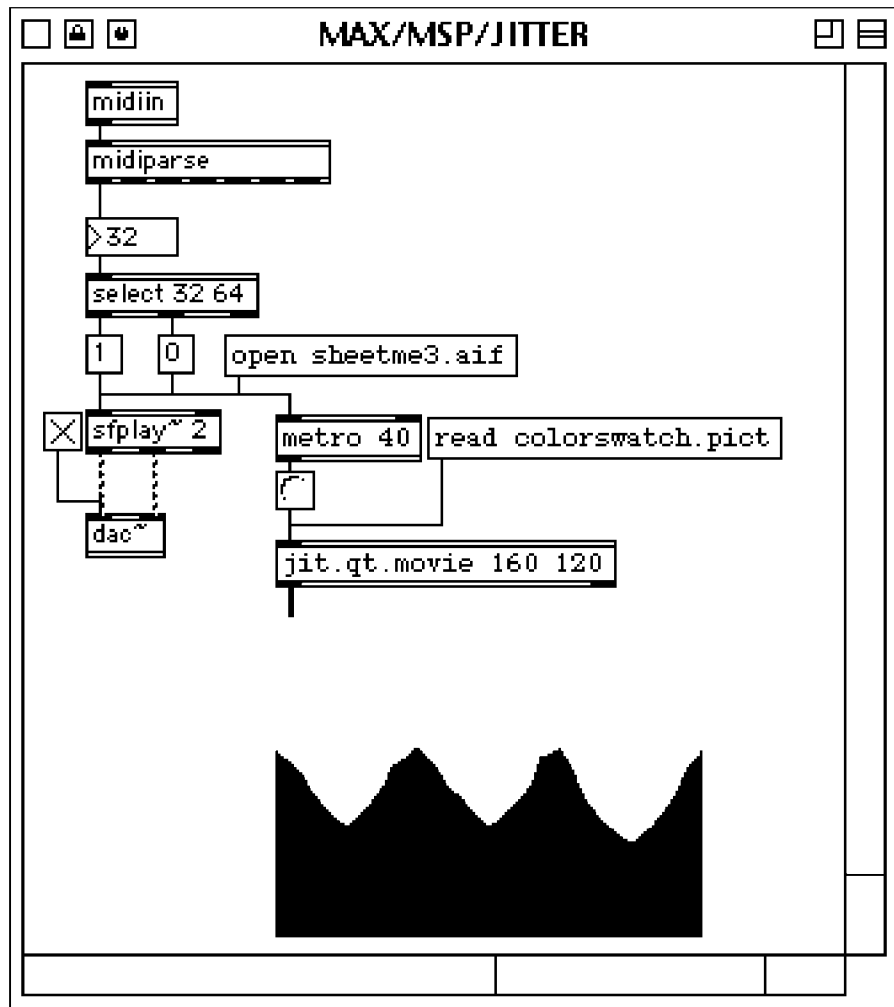


Figure 5

Figure 5 illustrates a basic Max program or “patch”, incorporating Max, MSP and Jitter objects. This patch accepts external MIDI data, as received from the M-SEA Interface, and uses the Max `midiparse` object to select MIDI note data. A Max number box then displays the incoming data. The Max `select` object selects the numbers 32 and 64, and passes these out the `select` object’s outlets. Here the number 32 will trigger a number 1 in the following

Max message object, and 64 will trigger a 0. The 1 is used to a) start playing an aiff file loaded into the MSP object `~sfplay`, and to b) display a colour swatch pict stored in the Jitter `jitter.qt.movie`. The aiff file output is through the MSP `~dac` object, and the pict file is displayed with a Jitter `jitter.pwindow`. The number 64, when received by `midin` is converted to a 0 and subsequently stops both the aiff and the pict files.

### 3.1 The M-SEA software

Figure 6 shows the current (OS9) M-SEA Interface, which is comprised of four main windows, labelled M-SEA, Monitor, Distances and Activity. The Scales window appears when the user selects the Scales Program from a Program Menu (not shown), and the Max window is shown only when the software is in the pre-packaged or development phase.

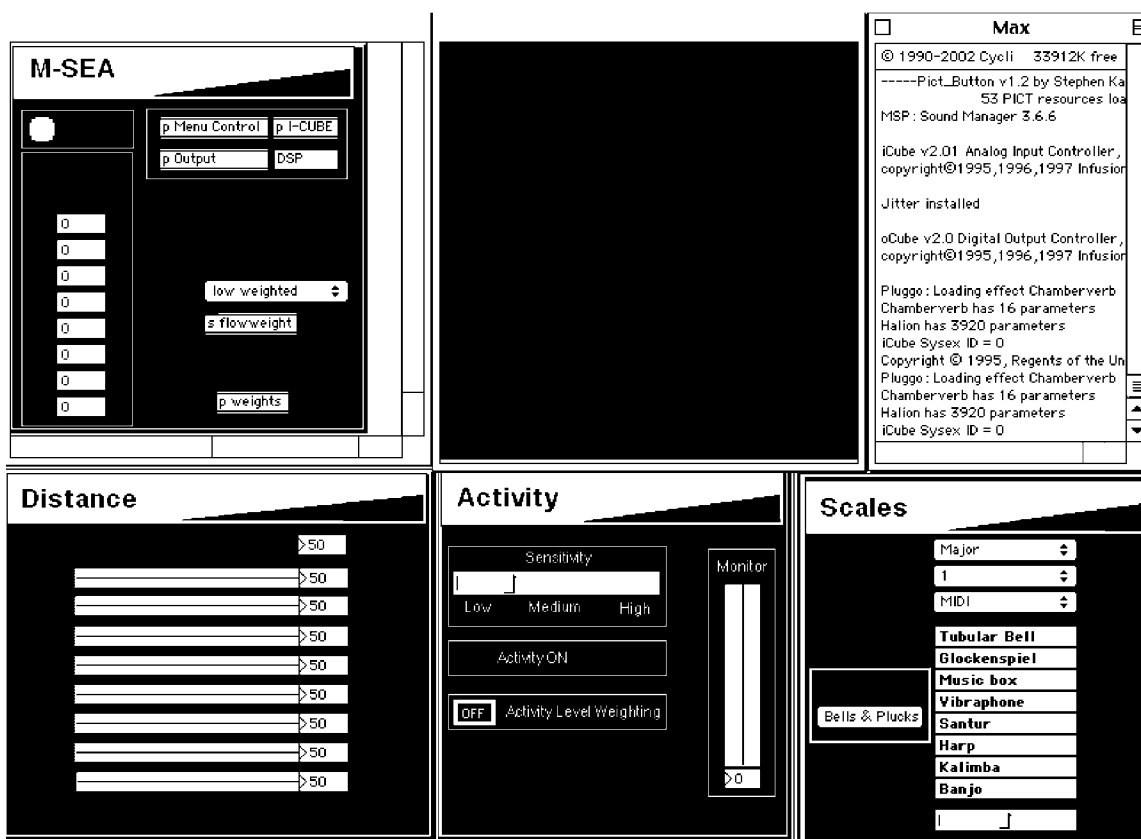


Figure 6

The M-SEA window is the main control area of the program. Here the incoming data (Input) from the eight sensors can be monitored, and the system reset to default settings. Controls for audio output are accessed with the Output object, in the form of soft sampling and signal processing plug-ins. The I-Cube object allows software control of the hardware interface, and the DSP button allows the user to select audio output (eg Sound Manager, ASIO Drivers etc.) and monitor CPU usage. The Menu Control object and Weighting objects are for development purposes only and will not appear in the packaged version.

The Monitor window provides a graphic representation of the sensing area. Movement within the space results in white dots appearing within the monitor window, these representing people or objects within the space.

The Distances window allows the configuration of the distance measurement to be used as incoming data from each of the eight sensors. In a 100sqm room the Distance measurement will be set to 64 (the number of milliseconds required for sensor emissions to be sent and returned to the sensor). An 80sqm room will require a setting of 51.

The Activity window provides control of the system when in Activity mode. In this mode incoming data from the sensing area is summed every two seconds and converted to an activity level on a scale of 1 – 10. The window provides a monitor of the activity level, a sensitivity selector with which the activity level can be altered to suit the number of people in the space, and an activity weighting button that activates a user-configurable weighting of the ten activity levels according to the amount of activity in the space.

#### 4. Boids

The accessibility of a 100sqm meter physical space for testing the M-SEA software, and in particular testing the music programs developed with Max, has proved to be a limitation during the development of the system. To overcome this, a virtual space has been incorporated into the M-SEA software, based on the Boids algorithm of Craig Reynolds (Reynolds, 1987).

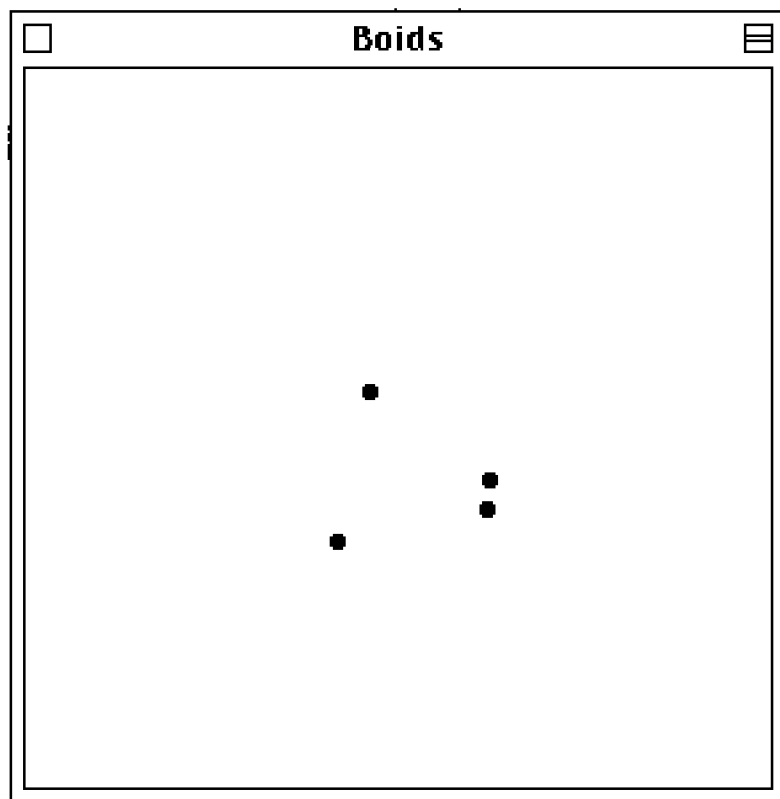


Figure 7

Boids simulate the flocking behaviour of birds and animals in an artificial environment, the Boids responding to various parameter settings including speed, attraction to one another, neighbours to which they will respond and so forth.

The Boids algorithm has been implemented as a Max object by Eric Singer who describes it as “a bird flight and animal flock simulator...based on the same algorithm which is used in Jurassic Park” (Singer 1995). Use of the Boids algorithm in M-SEA was pertinent as there may be any number of Boids simultaneously active within the artificial space and each Boid will exhibit behaviour according to where other Boids are located within the space. This is indeed similar to the number, movement and interactions of people within the physical M-SEA space. In the virtual space the Boids may move along any trajectory within their space without limitation, analogous to the unlimited movements of people within the M-SEA space.

Figure 7 shows a screen shot of the M-SEA Boids window with four Boids, represented as black dots within the space. When Boids are activated in the M-SEA software, the Boids window replaces the standard Monitor Window shown in Figure 6.

Further to the practical uses of the Boids as a simulation of the M-SEA sensing space, the Boids may also be used to generate musical materials in a controlled manner, via the manipulation of Boid parameters, and the manipulation of musical parameters in the M-SEA software. Such an application leads to the possibility of the software being used as a real-time music composition program, the results being recorded as composer/computer interactions, in the tradition of interactive music composition software (Chadabe, 1997, Rowe, 1993).

## 5. Music

There are currently eight of a projected ten default programs written for the M-SEA system. The default programs are essentially files that are opened in the M-SEA application. The default files can be called from the Program Menu of the application.

There are three approaches used for the mapping of sensor data to musical sounds within M-SEA. The first, and simplest, is Triggering, where specific points in the space are assigned to a sound or sequence of sounds. The sensors each have a maximum of 64 possible location points, however in this approach ten locations or less will be usual. As an example, a sensor may be divided into four triggering locations, as shown in Figure 8: location 1 (closest to the sensor) may trigger a bass drum, location 2 may trigger a piano chord, location 3 may trigger a flute melody and location 4 may trigger a sampled dog bark.

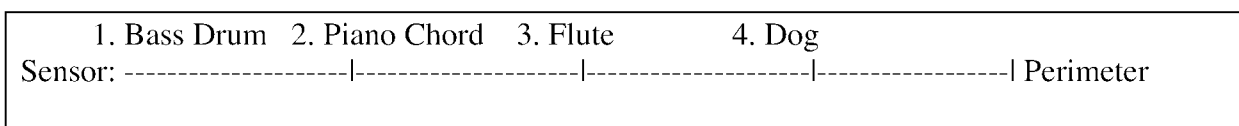


Figure 8

The second approach is to monitor Activity, the number of locations that have been sensed within a short time span (generally one to two seconds). Low activity levels may be assigned to a single instrument with a quiet slow moving melody, through to high activity levels where there are numerous instruments, rapid movement, multi-layers of melody and loud dynamic levels.



The third approach is a combination of the preceding two, a Hybrid approach where activity may control various musical parameters of a piece, triggers then controlling further parameters. Numerous combinations of the two approaches are possible, given the possibilities of assigning the sensor ranges into as many as 64 points or locations.

### ***Scales***

The simplest program available is a Triggering program entitled Scales. In this program each of the eight sensors is assigned eight notes from a musical scale, the highest note sounding when a person is detected close to the sensor, progressively lower notes occurring as the person moves further away. An analogy is to an eight-note piano keyboard that is expanded in size to reach from one side of the space to the other. Walking nearer and farther from the sensor is akin to walking up and down a large imaginary keyboard. As each sensor is assigned the same eight notes, but generally on different instrument sounds, the effect is to have eight large keyboards criss-crossing the room. The output of the program is MIDI-based, the user able to assign each of the eight sensors to any desired MIDI sound, either synthesised or sampled.

### ***Marimbas***

Marimbas is primarily based on marimba ostinati (repeated sound loops or patterns) that are supported with percussion and vibraphone melodies. The work is based on the triggering of four marimba ostinati (on sensors 1 – 4) this forming the underpinning of the piece, with four changes in harmony triggered on sensor 8. Sensor 5 provides control over dynamic levels from soft to loud and the vibraphone and further marimba melodies triggered on sensors 6 and 7. When the loudest dynamic level is active, further points on sensor 7 activate percussive loops.

### ***Harp & Percussion***

Another Triggering program, this simple musical environment is based on four adjacent sensors only, one controlling percussion loops, the remaining three controlling high, middle and low harp sounds respectively. Dynamic levels and rhythmic activity are controlled individually on each sensor, and harmony is controlled by the high harp sensor.

### ***Farmyard***

This program is designed for small children and relates, primarily through audio, a visit to a farmyard. As children move around the space they trigger the sounds of various farmyard animals, are followed around by, and give milk to a cat, hear dogs, feed chickens etc. Numerous video and still image projections can further the experience, and props can also be incorporated.

### ***Flow***

Flow, an Activity program, was composed for the Thuringowa River Festival in 2002. The theme for the festival was “Water for Life”, the M-SEA composition reflecting this directly through the use of water samples, and more indirectly, with vibraphone and marimba samples that represented the various flows of the river. These were augmented with a range of

percussion samples and a plucked piano sample that is used as a bass voice throughout the piece.

### ***Kafui***

A second Activity program, Kafui (pronounced caff-way) is an African drumming piece. Low activity levels provide sparse drumming patterns that increase in density with higher activity levels and are joined by more and more instruments. Probability tables control the selection of an array of patterns at high activity levels, resulting in a continually varying African drumming piece.

### ***Trance***

A Triggering piece, Trance uses a variety of loops, samples and sequences to form a three-movement dance piece. The dancers collectively become the D.J., controlling the music themselves. Based on a Euro-Trance/Euphoria style, the piece incorporates a variety of pads, arpeggios, simple harmonic changes, drum beats and bass loops to provide a variety of textures.

### ***Southern Lights***

A Hybrid approach is applied in this ambient style electronic work, in which two synthesis systems (Granular Synthesis and Scanned Synthesis) allow the user to manipulate synthesis parameters in real-time, thus creating their own sounds according to their location within the space. The sounds are complimented or contrasted with various triggered samples and sound-files. The Jitter environment is also incorporated into the work, the synthesis parameter controls also applied to simple images (a colour swatch and images of auroras), allowing the shifting of colours in the image. The image is projected through the space and the audio is output via four channels (quadraphonic). The panning of the audio over the four channels is controlled by those within the space.

## **6. Conclusion/Directions**

The development of the M-SEA system has to date represented a considerable personal voyage for the author. This voyage has incorporated the exploration of many facets of software and hardware development, numerous computer algorithms, such as that used for the Boids simulation of the space, and initial explorations of real-time projected image manipulations in the M-SEA space, as used in the Southern Lights program.

Presently the system and its software are stable, allowing for the development of the three remaining default programs, and further programs developed for specific situations. In August 2003 an intense period of development is to begin with the author undertaking a sabbatical to specifically explore applications of the M-SEA system. Planned work includes the development of programs specifically for disabled and special education workshops to take place in Sydney in December 2003, and the development of a work for Townsville's professional dance company, Dance North, in 2004. The second version of the hardware interface is to be completed, the software reconfigured for both the new hardware and Mac OSX. User-configurability is to be incorporated into the software, and finally, the software packaged as a stand-alone application, either for use with the hardware system, or without the hardware system in a virtual environment.

## References

- [1] CHADABE J. (1997) *Electric Sound: The Past and Promise of Electronic Music*. New Jersey: Prentice Hall.
- [2] CYCLING 74 (2003) *Max/MSP History and Background* [online]. Available from: <http://www.cycling74.com/support/questionsmsp.html> [Accessed 29<sup>th</sup> April 2004].
- [3] GARNCARZ D., VELA-MARTINEZ C., WALKER L. (1998) *I-Cube System Manual* Vancouver BC: Infusion Systems.
- [4] GEHLHAAR R. (1991) *SOUND=SPACE: An Interactive Musical Environment*. *Contemporary Music Review* **6** (1) 59-72.
- [5] GEHLHAAR R. (1998) *SOUND=SPACE Workshops for Disabled Children* [online]. Available from <http://www.gehlhaar.org/resfelr.htm> [Accessed 29<sup>th</sup> April 2004].
- [6] POLAROID OEM COMPONENTS GROUP (1999) *Technical Specifications for 6500 Series Sonar Ranging Module* [online]. Available from: <http://www.polaroidoem.com/pdfp6500series.pdf> [Accessed 29<sup>th</sup> April 2004].
- [7] REYNOLDS C. (1987) *Flocks, Herds and Schools, A Distributed Behavioural Model* *Computer Graphics* **21** (4) 25-34.
- [8] ROWE R. (1993) *Interactive Music Systems* Massachusetts: The MIT Press.
- [9] SINGER E. (1999) *Boids v1.51 Help File* [online]. Available from <http://www.ericssinger.com/Downloads/Boids.1.51.sea.hqx> [Accessed 29<sup>th</sup> April 2004].
- [10] SUGA N. (1990) *Bisonar and Neural Computation in Bats* *Scientific American* **253** 60-68.