LISTENING TO THE MIND LISTENING ENTRY: PROPOSED FOR INCLUSION IN ICAD 2004, SYDNEY OPERA HOUSE, 6 – 9 JULY 2004

"THE LIVING MIRROR"

Matthew Sanderson

email: modgeulator@yahoo.com.au

ABSTRACT

My aims with this sonification titled "The Living Mirror," are as follows:

- To create organic sounds. I wanted the listener to be able to tell that a living organism was controlling the sounds.
- To create a lot of space in the sound field while still having a very detailed sonic picture. To make use of almost all the data without the piece becoming cluttered.
- To find a balance between music and sound. To find naturally occurring rhythmic and musical structures in the data.
- To use 3D positioning to create a more detailed sonic image than would be possible in stereo.

SOUND FILES

File Name	Spatial Location
Sanderson01.wav	Fp1
Sanderson02.wav	Fp2
Sanderson03.wav	F7
Sanderson04.wav	F3
Sanderson05.wav	Fz
Sanderson06.wav	F4
Sanderson07.wav	F8
Sanderson08.wav	FC3
Sanderson09.wav	FCz
Sanderson10.wav	FC4
Sanderson11.wav	T3
Sanderson12.wav	C3
Sanderson13.wav	Cz
Sanderson14.wav	C4
Sanderson15.wav	T4
Sanderson16.wav	CP3
Sanderson17.wav	CPz
Sanderson18.wav	CP4
Sanderson19.wav	T5
Sanderson20.wav	P3
Sanderson21.wav	Pz
Sanderson22.wav	P4
Sanderson23.wav	T6
Sanderson24.wav	O1
Sanderson25.wav	Oz
Sanderson26.wav	O2
Sanderson27.wav	Front Centre Floor (1.0, 0, 0)
Sanderson28.wav	Back Centre Floor (1.0, 180, 0)

MAPPING

The ECG text files are first rescaled to 32 bit integers in the range of 0 to 1. From this data the following mappings can be made for each channel:

ch01-Fp1, ch02-Fp2 (Forehead Left, Forehead Right)

- Reduce data to 15,000 samples. For every ten samples find an average and a min/max range.
- Map the averages to MIDI notes by multiplying all numbers by 127 and rounding to integer values. Map these averages into a C Major key.
- Map the range min/max values to velocity by multiplying by 127 and rounding to integers.
- Find the offset times of the 400 largest range min/max values and trigger a corresponding note and velocity at each of these times.
- Use a timbre with a fast attack and short release time.

ch03 to ch26 (The other 24 sensor electrodes)

- Use the resulting MIDI data from ch01 and ch02 as a basis. Alter the timbre to have a longer attack and a much longer decay time. Each channel uses both instrument sounds (ch01 and ch02).
- Reduce the data for each channel to 15,000 samples. For every ten samples find an average and a range min/max.
- Map the averages to pitch bend events by multiplying by 127 and rounding to integers. Use a pitch bend range of +-6 semitones.
- Find the offset times of the 40 largest min/max range values to trigger an amplitude change. The amplitude is set at a value of 7 until one of these offset times is reached at which time it is set momentarily to 127.
- Use this additional MIDI data in conjunction with the MIDI data for ch01 and ch02.
- The entire sound has a reverb effect applied to it.
- The mix of each channel's two instrument sounds (based on ch01 and ch02) is determined by its left/right position in relation to ch01 and ch02 (left and right forehead sensors). So, for example, channels based on sensors on the left side of the head would have more signal from ch01. Emphasis is placed on creating a widened audio image rather than an accurate portrayal, so the sounds are mapped as follows:

Channel	ch01 amount	ch02 amount
ch03-F7	100%	0%
ch04-F3	80%	20%
ch05-Fz	50%	50%
ch06-F4	20%	80%
ch07-F8	0%	100%
ch08-FC3	100%	0%
ch09-FCz	50%	50%
ch10-FC4	0%	100%
ch11-T3	100%	0%
ch12-C3	80%	20%
ch13-Cz	50%	50%
ch14-C4	20%	80%
ch15-T4	0%	100%
ch16-CP3	100%	0%
ch17-CPz	50%	50%
ch18-CP4	0%	100%
ch19-T5	100%	0%
ch20-P3	80%	20%
ch21-Pz	50%	50%
ch22-P4	20%	80%
ch23-T6	0%	100%
ch24-O1	100%	0%
ch25-Oz	50%	50%
ch26-O2	0%	100%

ch27 to ch30 (Sensors around eye)

[UNUSED]

ch31-Erbs (Point references heart rate)

- Reduce data to 15,000 samples. For every ten samples find an average and a min/max range.
- Multiply numbers by 127 and round to integers. Averages are mapped to a C major key and assigned to notes.
- Min/max values are mapped to velocity.

ch32-OrbOcc (Orbicularis Occuli)

- Reduce data to 15,000 samples. For every ten samples find an average, a smallest value and a min/max range.
- Find the offset time of the smallest value (it's at 8,309 of 15,000). From this offset find the 40 largest min/max sizes.
- Map the averages to pitch bend events by multiplying each number by 127 then rounding to integers.
- At the offset times of each of the 40 largest values sound a note at MIDI note 66 (F#, octave 6) using a trumpet sample.

ch33-MassMasseter (Jaw muscle – jaw clenching)

- Reduce data to 15,000 samples. For every ten samples find a min/max range.
- Find the largest min/max range (it's at 2,799 of 15,000). From one sample after this offset time re-normalize all min/max size values to a 0 to 1 range, multiply by 127 and round to integers. Use this data to control amplitude and a filter cut-off.

ch34-EDA (Electrodermal activity – sweat response)

[UNUSED]

ch35-RespBreathing (Breathing)

- Reduce data to 15,000 samples. For every ten samples find a smallest value.
- At the offset time of the single smallest value in the data (4777 of 15,000) play a loud bang sound.

ch36-ECG (Heart rate)

- Reduce data to 15,000 samples. For every ten samples find an average, a smallest value, a largest value and a min/max range.
- This sonification uses two different synthesized instrument sounds and a phase vocoder.
- For the first sound, map the largest values to notes by multiplying all values by 127 and rounding to integers. Map the min/max sizes to velocity by multiplying all values by 127 and rounding to integers.
- For the second sound, map the smallest values to notes by multiplying all values by 127 and rounding to integers, then round to fit a C major key. Invert the min/max size values then map to velocity by multiplying all values by 127 and rounding to integers.
- The first instrument sound is sent to a phase vocoder as the modulator. The second instrument is used as the carrier. Averages are mapped to a "shift" parameter on the vocoder by multiplying all values by 127 and rounding to integers.

ACKNOWLEDGEMENTS

Thanks to David Worrall for advice, assistance and initial file conversions.