

El gesto composicional en mi obra reciente.

Luis Tamagnini

Apuntes sobre la aplicación de Computación Musical en la organización de la altura, el ritmo, la síntesis y el procesamiento de sonido, la espacialización y la formalización de las estructuras musicales.

Introducción.

En el siguiente texto voy a exponer aspectos relativos a la composición de mis cuatro últimas piezas: *Escenas de la vida material*, para piano, ensamble y electrónica en vivo; *Pris*, música electrónica; *A/V*, performance electrónica audiovisual; y *Caverna*, para marimba, vibráfono y electrónica audiovisual en vivo. Estas piezas fueron compuestas durante un período de un año. *Escenas...* fue escrita para un Ensamble de Improvisación creado durante un taller realizado en la Escuela de Música de la UNR en 2013, y estrenada en un concierto en la misma escuela. *Pris* fue estrenada en el concierto Pulsos #5 en el CC Parque de España de Rosario en 2013.

Voy a referirme a los criterios aplicados en la composición de las mismas en relación a la organización de la altura, el ritmo, el material electrónico, la forma, la utilización del espacio, la interpretación, y la utilización de herramientas de computación musical, entre otros. Asimismo voy a apuntar algunas ideas técnicas y estéticas de mi pensamiento musical actual.

La altura.

Una de las posibilidades más atractivas de la computación musical es sin dudas la experimentación en la organización de las alturas. La historia de la música nos provee de una extensa colección de técnicas y procedimientos más o menos automáticos o algorítmicos. Por otro lado, las herramientas computacionales nos sirven para re-construir, combinar e intervenir esas técnicas en programas de composición automática, instrumentos musicales electrónicos o piezas definidas. El uso de la potencia computacional para este fin no es nueva y constituye un de los ejes fundamentales del desarrollo de la computación musical durante la segunda mitad del SXX. Además, la

herramienta funciona como nexo entre la sustancia puramente electro-acústica, la materia sonora; y el pensamiento tradicional de control de la altura; y abre el juego a la exploración de un sinfín de estados intermedios y modulaciones.

Mi primer experiencia en este campo data del año 2008 y fue desarrollada en Max/MSP para una pieza audiovisual electrónica llamada Rubick. Para este trabajo desarrollé un patch de serialismo dodecafónico, con control preciso de varios aspectos como la serie por supuesto, las transposiciones de la serie, y la registración, entre otros. Mas adelante, incorpore la posibilidad de modificar todos estos parámetros en tiempo real y descubrí que había creado un instrumento. La pieza deviene instrumento musical. Este patch fue luego utilizado en muchos otros proyectos hasta el día de hoy debido a la flexibilidad y capacidad de generación de estructuras musicales que tiene.

The image shows a complex Max/MSP patch interface. At the top left, there are control parameters: 'MSP START' (checkbox), '78 BPM', '1. GATE', and 'REG' (value 9). Below this is an 'init settings' section with a grid of buttons and checkboxes for 'REG FIX / CHANGE', 'ROW FIX / CHANGE', and 'USE INV'. The 'MIDI SOURCE' section includes 'IAC Driver Bus 1' and 'open AudioSetup'. The 'MAIN OUT' section has five vertical sliders and a 'SYNTH PATCH' button. A 'seq' window shows three sequencer lanes with 'forward' and 'random' buttons. The central area contains a grid of colored boxes (pink, green, blue) representing musical modes and intervals, with a list of names on the right: Maj7, 7, 7b5, Maj7b5, 7dimb5, m7Maj, m7, m7b5, 7dim, m7dim, mMaj7b5, +Maj7, +7, jónico O-7, T dórico O-7, frigio O-7, and lidio. At the bottom, there is a 'registorador' window with a 'TRIGGER REG' button and a keyboard interface. The right side of the patch has 'MODO X' (value 0) and 'MODO Y' (value 0) controls, along with another grid of buttons and checkboxes for 'REG FIX / CHANGE', 'ROW FIX / CHANGE', and 'USE INV'. A 'LOGIC SCRIPTS' window is also visible.

Fig 1 – Patch de Rubick.



Fig 2 – Detalle donde puede apreciarse el ajuste para implementar el control gestual externo a través de MIDI.

Una consecuencia directa de esto fue el uso de este patch como banco de pruebas para diferentes situaciones armónicas. Esto afirma otro de los postulados de la computación musical para mi desconocido en ese entonces, el uso de la misma como soporte para la composición, es decir, un medio en el cual apuntar gestos composicionales y generar rápidamente muchas derivaciones del mismo, lo que se conoce como CAC, Composición Asistida por Computadora. En este laboratorio armónico elaboré un concepto que llamo "multiplicación serial" y que utilizo en *Pris*.

Pris es una pieza de música electrónica, octofónica y generativa. Es decir, todos los sonidos son sintetizados en tiempo real y espacializados en ocho canales por algoritmos que representan un modelo lógico de la pieza. Algunos parámetros son controlados en forma estricta mientras que otros asumen cierto grado de aleatoriedad, lo cual le da a la pieza su carácter generativo. Ninguna "interpretación" de *Pris* es exactamente igual. La pieza fue totalmente desarrollada en SuperCollier.

El principio de multiplicación serial está derivado de la técnica dodecafónica y consiste en, dados dos conjuntos de alturas, un conjunto A o serie principal, y un conjunto B o serie de transposición; generar progresiones y/o simultaneidades de alturas a partir de ir presentado en sucesión el conjunto A transpuesto a cada altura establecida por el conjunto B.

En *Pris* existen solo dos series que estructuran toda la pieza, un conjunto simétrico de seis sonidos constituido por la sucesión de un tritono y un tono; y un conjunto de solo dos notas; un intervalo de quinta justa.

```
~perfect = [0,7];
```

```
~sym62 = [11,5,7,1,3,9];
```

A lo largo de la pieza se utilizan ambas series en función principal y transposición, y las retrogradaciones de las mismas de la misma manera; obteniéndose así cinco series derivadas de doce notas.

```

~sym62 +.x ~perfect
~perfect +.x ~sym62
~sym62.reverse +.x ~perfect
~perfect +.x ~sym62.reverse
~sym62.reverse +.x ~perfect.reverse

[ 11, 6, 5, 0, 7, 2, 1, 8, 3, 10, 9, 4 ]

[ 11, 5, 7, 1, 3, 9, 6, 0, 2, 8, 10, 4 ]

[ 9, 4, 3, 10, 1, 8, 7, 2, 5, 0, 11, 6 ]

[ 9, 3, 1, 7, 5, 11, 4, 10, 8, 2, 0, 6 ]

[ 4, 9, 10, 3, 8, 1, 2, 7, 0, 5, 6, 11 ]

```

Si transportamos a 0 estas series observamos que las cinco son diferentes, es decir no existe relación directa entre ninguna de ellas, por transposición, retrogradación o inversión.

Es curioso notar aquí que la multiplicación de las series origina conjuntos cuya longitud es igual a la longitud de la serie principal por la longitud de la serie de transposición, en este caso dos por seis es igual a doce. Además, en este caso, la naturaleza de las series origina conjuntos sin alturas repetidas, es decir, cinco diferentes series dodecafónicas. Esto no fue previsto de antemano, solo los dos modos o series "generadoras".

Por otro lado, la elección de estos dos conjuntos primarios no es arbitraria y obedece a una búsqueda recurrente en mi música; el inter-juego de la consonancia con el atonalismo. La utopía de un atonalismo consonante, o al menos con un alto grado de consonancia. Esto puede verse en las series resultantes en donde las quintas justas (o cuartas justas) aparecen a veces mas explícitamente, otras veces mas enmascaradas entre tritonos y segundas mayores. También aparecen en menor medida algunos otros intervalos, el semitono y la tercera menor.

El ritmo.

Otro de los aspectos en donde la computación musical puede aportar muchas herramientas e ideas es indudablemente el ritmo.

En *Pris*, el material rítmico es generado a partir de dos estructuras iniciales, mas dos variantes.

```
~rA = [ 1, 1, 0.2, 1, 1, 0.2, 1 ];
~rB = [ 1, 0.2, 1, 1, 0.2, 1, 1, 1 ];
~rA2 = [ 1, 1, 0.33, 1, 1, 0.33, 1 ];
~rB2 = [ 1, 0.33, 1, 1, 0.33, 1, 1, 1 ];
```

Es visible en el código la claridad que conlleva la representación de las estructuras rítmicas con valores numéricos. Podemos observar que el modelo es análogo al que utilizamos para representar la altura, por lo que también podemos intuir que procedimientos similares pueden ser aplicados sobre los mismos. A lo largo de la pieza, patrones rítmico-melódicos se van formalizando a partir de la combinación de las series de alturas antes mencionadas con variaciones de estas estructuras rítmicas. Estas variantes incluyen a menudo elementos estocásticos. Aquí la aleatoriedad es introducida en un nivel micro-formal, de esta manera la estructura mantiene buena parte de su cualidad perceptiva.

Los procedimientos aplicados a las estructuras son sencillos, e incluyen: la división por un valor fijo o aleatorio para generar una aumentación o disminución, y la permutación libre de los elementos de la estructura, sin alterar el número de componentes. Esto modifica la posición de los valores cortos en relación a los valores largos.

```
~rB2/0.5; >> [ 2, 0.66, 2, 2, 0.66, 2, 2, 2 ]
~rB/rrand(1.7,2.0); >> [ 0.5436986535943, 0.10873973071886, (...)]
~rA.scramble; >> [ 1, 1, 0.2, 1, 1, 0.2, 1 ]
~rB.scramble.pow(1.25); >>[1, 1, 0.13374806099528, 1, 1, 1, 1, 0.13374806099528]
```

En el último caso citado, la potenciación genera un acortamiento dramático de los valores cortos, que asumen así el rol de apoyaturas o notas ornamentales.

Más estocástica.

A/V es una pieza electrónica audiovisual generada en tiempo real e interpretada, es decir, no existe partitura ni versión fija de la misma (mas allá de los registros de concierto). El intérprete puede jugar libremente con una serie de estructuras musicales generativas que puede combinar a su gusto. Estas estructuras son patrones rítmicos de longitud indeterminada (o determinada por el intérprete) de sonidos sintéticos,. En la generación de estos patrones se aplica una buena cantidad de aleatoriedad y en el caso del ritmo por ejemplo utilizo funciones como la que se detalla a continuación,

```
scramble( {1/Array.geom(3, 4, 2).choose} !16);
```

y un posible resultado.

```
[ 0.125, 0.0625, 0.25, 0.0625, 0.25, 0.25, 0.125, 0.125, 0.25, 0.0625, 0.125,
0.0625, 0.125, 0.0625, 0.25, 0.125 ]
```

África.

Caverna, la mas reciente de las cuatro piezas que describo en este ensayo, es para marimba, vibráfono y electrónica en tiempo real. La electrónica incluye tanto sonidos electrónicos sintéticos como el procesamiento de los instrumentos acústicos utilizados. Además, también se genera en tiempo real una componente de vídeo en sincronía con los sonidos electrónicos.

Caverna está basada en el "Mito de la *Caverna*" de Platón y utiliza materiales de un ritmo originario de África Occidental llamado Abioweka. Este ritmo es usado en una danza para niños, hecho que da origen al nombre del segundo movimiento de la pieza, donde aparece el material citado, "Danza de los niños".

En la práctica original africana, se utilizan un número no determinado de djembés y tambores de distintos tamaños.(dunum, sangban, kenkeni) Cada uno de ellos toca uno de una serie de patrones rítmicos. Un djembé "solista" oficia de director marcando con un patrón especial, la "llamada", articulaciones entre solos y tuttis. En estas articulaciones se pueden intercambiar los participantes de la danza y los ejecutantes pueden comenzar a

tocar su parte o descansar. Así, los diferentes patrones se van superponiendo en una estructura poli-rítmica de mucha riqueza.

The figure displays musical notation for seven different instruments, all in a 6/8 time signature. The notation uses 'x' for a specific sound and various rhythmic symbols (vertical lines, beams, and flags) to indicate timing and duration. The instruments and their patterns are:

- Djembé - Llamada:** A pattern of six notes, each marked with an 'x', followed by a rest.
- Djembé - Pattern 1:** A pattern of six notes, each marked with an 'x', followed by a rest.
- Djembé - Pattern 2:** A pattern of six notes, each marked with an 'x', followed by a rest.
- Djembé - Pattern 3:** A pattern of six notes, each marked with an 'x', followed by a rest.
- Djembé - Pattern 4:** A pattern of six notes, each marked with an 'x', followed by a rest.
- Dunum (Tambor bajo):** A pattern of six notes, each marked with an 'x', followed by a rest.
- Sangban (Tambor tenor):** A pattern of six notes, each marked with an 'x', followed by a rest.
- Kenkení (Tambor alto):** A pattern of six notes, each marked with an 'x', followed by a rest.
- Campana o cencerro:** A pattern of six notes, each marked with an 'x', followed by a rest.

Fig 3 - Abioweka

En *Caverna*, los patrones rítmicos de la Abioweka se utilizan sin modificar, casi de la misma manera que en la práctica original. Una serie de sonidos sintéticos de djembé y tambores son organizados por patrones que son disparados en vivo por el director o realizador de la electrónica. Sobre esta estructura los instrumentos ejecutan una partitura abierta de patrones rítmico-melódicos derivados del material de la Abioweka. La única diferencia sustancial respecto de la práctica original es la posibilidad de superponer los patrones sobre cualquier tiempo de la métrica, aumentando las posibilidades combinatorias de las estructuras poli-rítmicas.

En cuanto a la resolución técnica, utilizo aquí un secuenciador programado en SuperCollider con Patterns, una de sus herramientas más poderosas, y controlado a partir de un controlador midi mediante interruptores sencillos. Cada interruptor determina la ejecución o no de un patrón rítmico.

```
~clock = TempoClock.new(110/60,4);

~abioueka_call = Pbind (
    \instrument, \dj_tone,
    \dur, Pseq([0.11,0.22,0.33,0.33,0.33,0.69,0.33,0.69,0.97],1),
    \amp, Pseq([0.8,1,0.8,0.8,0.8,1,0.8,1,1]*0.5,1),
    \out, ~djembe_tone,
    \finish, Pfunc {|ev| ~host.sendMessage("/abi",1)}
).play(~clock);
~abioueka_call.stop;
```

Los Patterns son estructuras de datos que almacenan parejas de \tags o palabras de referencia y datos o secuencias de datos. Estas estructuras pueden controlarse con relojes y enviar de manera secuencial estos datos a algoritmos de síntesis o a otros dispositivos a través de MIDI u OSC (Open Sound Control). Constituyen una de las herramientas principales para la organización formal de las estructuras musicales en SuperCollider.

El diseño del sonido electrónico.

El campo de la síntesis es y ha sido desde los inicios de la computación musical en la década del 50, el otro pilar principal sobre el que se sostiene el desarrollo de esta disciplina. Comúnmente llamada CGM, Computer Generated Music; quizá sea esta la cara más visible (o audible) de la computación musical, y la más practicada en la actualidad, gracias al desarrollo que ha experimentado en sincronía con el aumento del potencial de cálculo de las máquinas.

En mis trabajos he utilizado procesos tradicionales de síntesis y otros en los que se introducen elementos estocásticos en varios niveles.

En primer lugar, para citar el primero de los casos voy a referirme al diseño de un sonido de Djembé, para la pieza *Caverna*. El Djembé es un tambor de origen africano, el instrumento mas importante de la cultura de África Occidental y su nombre significa y simboliza la paz. Es un tambor en forma de copa, tallado en un tronco de madera dura y tiene un parche de cuero atado. Se pueden obtener tres sonidos básicos percutiendo el parche de diferentes maneras, el bajo, el tono y el slap. Para recrear estos sonidos escribí algoritmos de síntesis aditiva siguiendo el modelo espectral de los mismos.

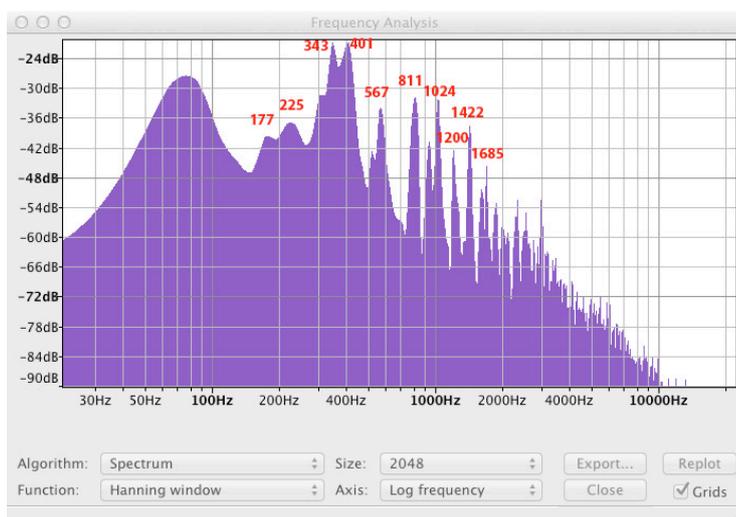


Fig 4 – Espectrograma del tono de Djembé.

A continuación el código en SuperCollider para recrear el sonido.

```
// tono
SynthDef(\dj_tone, { |out = 0, amp = 1, attack = 0.005, release = 0.35|
    var signal, fm, noise, env, env2, env3;
    env = EnvGen.ar(Env.perc(attack,release), doneAction:2);
    env2 = EnvGen.ar(Env.perc(attack,release/4));
    env3 = EnvGen.ar(Env.perc(0,release/8));
    signal =
Mix.ar(SinOsc.ar([75,177,225,343,401,567,811,1024,1200,1422,1685],0,[-33.dbamp,-
40.dbamp,-37.dbamp,-18.dbamp,-18.dbamp,-34.dbamp,-31.dbamp,-31.dbamp,-42.dbamp,-
37.dbamp,-45.dbamp])) * env;
    fm = SinOsc.ar(343 + SinOsc.ar(401,0,-18.dbamp),0,-18.dbamp) * env2;
    noise = LPF.ar(WhiteNoise.ar(0.05),2500) * env3;
    Out.ar(out, signal*2);
}).add;
```

Este algoritmo es esencialmente una síntesis aditiva, mas una FM simple y un poco de ruido para simular el ataque y la energía presente entre las formantes principales del espectro.

Procedimientos mas experimentales son aplicados en *A/V*. Esta pieza, que llamé “performance electrónica audiovisual” fue pensada desde la experimentación en todos sus aspectos, y es para mi una suerte de estudio y puesta a prueba de recursos técnicos. En ella, una serie de estructuras musicales generativas son controladas en vivo por un intérprete. El control supone la organización formal, es decir, la presencia o no de la estructura y además la modificación de parámetros o aspectos del sonido. Una de estas estructuras es un “drone”, es decir, un sonido sostenido indefinidamente generado a partir de modular en frecuencia, dos diferentes ondas portadoras, con una única onda moduladora. La doble FM resultante está espacializada en forma discreta, una en cada canal; la pieza es stereo. Luego, el intérprete puede modificar las frecuencias de los tres osciladores, y un índice de modulación general. También puede alterar todos estos parámetros en forma simultanea y aleatoria con un trigger o botón de disparo. Esto genera la percepción del timbre en una dimensión espacial, un timbre con cualidades formales, o temporales coincidentes, pero espectralmente divergentes en cada canal. He llamado a este concepto “espacialización orgánica”. A continuación se detalla el sencillo algoritmo utilizado en este caso.

```
// drone FM
SynthDef(\fmDrone, { |carrierFreq= # [100,100], carrierLvl=0.5,
modulatorRatio=5, modulatorIndex=1.5, outputAmp=0.2, sig, out=0|
  // simple FM core
  sig = SinOsc.ar(carrierFreq, 1, 0.5, 0.5) * carrierLvl;
  sig = sig + SinOsc.ar(carrierFreq * modulatorRatio) * modulatorIndex;
  sig = cos( sig * 2pi) * outputAmp * 0.06;
  Out.ar(out, sig);
}, [0.2,0.2,0.2,0.2,1]).add;
```

En *Pris*, los procesos estocásticos son aplicados tanto en la generación de los sonidos como en la organización formal de los mismos. Primeramente, un conjunto de 100 sonidos percusivos y otros 100 sonidos continuantes, son generados a partir de un único

algoritmo complejo y que admite una muy alta parametrización. Luego, en las diferentes secciones de la pieza se utilizan subsets de estos timbres generados aleatoriamente por otro algoritmo. De esta manera se mantiene el carácter general de la pieza, al tiempo que el resultado tímbrico es siempre diferente.

La extensión del gesto instrumental. Procesamiento de instrumentos mecánico acústicos.

El procesamiento en tiempo real de instrumentos mecánico acústicos es otro de los recursos que empleo en alguna de mis piezas. La computación musical provee enormes capacidades para configurar sistemas de procesamiento específicamente diseñados y adaptados a los requerimientos técnicos y estéticos de la composición. En el caso de mi música, me interesa usar este recurso de manera que el sonido del instrumento y el sonido electrónico se manifiesten como un continuo sin fisuras, como una unidad perceptivamente indisoluble. Por otro lado, el software altamente optimizado, como es el caso de SuperCollider, minimiza los riesgos que conlleva la utilización de este recurso en vivo, al tiempo que concentra el potencial computacional en el problema que se intenta resolver aumentando dramáticamente las posibilidades técnicas y expresivas.

Escenas de la vida material es una pieza compuesta para un ensamble de improvisación que se formó en un taller llevado a cabo en la Escuela de Música de la UNR en 2013. En ella utilizo un piano procesado en tiempo real con diferentes algoritmos para las distintas secciones. El pianista además controla estos algoritmos con un dispositivo MIDI. En el inicio de la pieza el piano ejecuta un acorde que sostiene hasta su extinción acústica. Este gesto sirve además de marca de comienzo para el resto de los músicos. Durante un tiempo inmediatamente posterior al ataque del acorde, la computadora registra un fragmento del sonido, que luego es usado como material para un algoritmo de síntesis granular. Mientras el sonido acústico se extingue, el granulador comienza a actuar, generando la sensación perceptiva de sostenimiento indefinido del sonido del acorde inicial. Este gesto continúa durante toda la primera sección, aunque luego el pianista interviene modificando parámetros del granulador de manera que la cualidad electrónica del sonido se hace mas audible. En el algoritmo que se detalla mas adelante, puede verse además que el granulador puede ser encadenado a otros procesadores, todos stereo, que incluyen: un retardo doble, una cámara de reverberación, un banco de filtros

resonantes y un reductor de bits o distorsión digital. Estos procesos son utilizados en otras secciones de la pieza.

```
// granulador
SynthDef(\granular, {
|out = 0, buf = 0, amp = 1, trate = 50, factorDur = 12, pos =0.5, rate =1,
gate = 1, dlybuf, dly = 0, revbuf, rev = 0,crbuf, crush = 0, filbuf, fil = 0|
    var pan, dur, clk, center, signal, env;

    dur = factorDur / trate;
    clk = Impulse.kr(trate);
    center = pos * BufDur.kr(buf) + TRand.kr(0, 0.01, clk);
    pan = WhiteNoise.kr(0.6);
    env = EnvGate.new(gate:gate, doneAction:1);
    signal = Limiter.ar(TGrains.ar(2, clk, buf, rate, center, dur, pan,1) *
env,0.95);
    Out.ar(dlybuf,signal * dly);
    Out.ar(revbuf,signal * rev);
    Out.ar(crbuf,signal * crush);
    Out.ar(filbuf,signal * fil);
    Out.ar(out,signal * amp);
}).add;
```

En *Caverna*, también utilizo procesamiento en tiempo real en algunas secciones. En la primera sección, dos granuladores se aplican de manera similar a la descrita anteriormente. En la tercera sección la señal de entrada de los instrumentos, marimba y vibráfono es procesada por un algoritmo que realiza una simple multiplicación escalar de las mismas. Esto genera la ausencia de sonido si la señal en alguna de las entradas es nula o muy baja, pero produce un efecto similar a un modulador en anillo cuando ambos instrumentos están sonando. La presencia de este efecto en la electrónica es controlada por el director o realizador con un simple control de volumen. En esta misma sección también se utiliza la grabación de una extensa parte de los instrumentos, unos quince segundos. Estas grabaciones se reproducen luego a la mitad de su velocidad y de atrás para adelante, a la manera de un doble canon por aumentación y movimiento contrario. A continuación se detalla el algoritmo de modulación cruzada descrito anteriormente.

```

SynthDef(\crossRing, {|inA = 8, inB = 9, noise = 1, gain = 1, amp = 1, out = 0,
gate = 1|
  var signal, mod, env;
  env = EnvGate.new(gate:gate, doneAction:1);
  signal = In.ar(inA,1) * In.ar(inB,1) * gain;
  mod = LFNoise2.ar(noise)*3;
  Out.ar(out, Pan2.ar(signal, mod) * amp);
}).add;

```

El espacio.

El espacio siempre ha sido una preocupación excluyente en la música electrónica y electroacústica. En lo que a mi respecta nunca me he sentido atraído por los modelos tradicionales de simulación de espacios acústicos y diseño de trayectorias de movimiento de los sonidos. Las estrategias y conceptos propios de la computación musical me han sugerido otros acercamientos. A continuación voy a detallar algunos aspectos sobresalientes de los mismos.

El primero de ellos ya fue mencionado en el presente ensayo y tiene que ver con lo que yo llamo “espacialización orgánica”, en donde el espectro de un timbre es distribuido en dos o más canales. En estos casos no es el timbre el que se mueve alrededor del oyente, sino que es el oyente el que está inmerso dentro de un timbre en movimiento, movimiento espectral claro. Esta técnica fue utilizada en *A/V*, en el “drone” FM que se detalla más arriba.

Otra de las estrategias aplicadas es, como no podía ser de otra manera, la aleatoriedad. La localización puntual de un sonido en un lugar del espacio acústico definido estocásticamente. En *Pris*, la abundancia y hasta la saturación de sonidos percusivos muy cortos son localizados por funciones aleatorias en uno, de ocho canales o en un punto situado exactamente entre dos de ellos. Esto determina 16 puntos de localización posibles, todos situados en el perímetro del espacio acústico definido por el sistema octofónico, generando un efecto de caleidoscopio acústico. Esta técnica también es utilizada en *A/V* para espacializar en dos canales secciones de sonidos de percusión.

Finalmente, un acercamiento que remite a la utilización del espacio en el Renacimiento, he utilizado en *Caverna*, una pieza para tres canales compuesta para ser estrenada en el Espacio Cultural Universitario ECU de Rosario. Este edificio tiene una estructura sumamente grande y reverberante similar a una gran catedral. En la pieza, los canales uno y dos son situados al frente de la escena en una configuración stereo normal. Un tercer canal es colocado detrás de escena, totalmente afuera del espacio de concierto. Por este canal se reproducen los sonidos y ritmos que citan la Abioweka, el ritmo africano que utilizo en la pieza. La cita debe oírse como proveniente de un lugar remoto.

Referencias.

Pris – registro sonoro <http://luistamagnini.bandcamp.com/album/pris>

Escenas de la vida material – registro sonoro <http://luistamagnini.bandcamp.com>

A/V – registro sonoro https://www.youtube.com/watch?v=c_M3wOWSaQQ

Caverna – registro sonoro <https://www.youtube.com/watch?v=Y7IJKNiBWeE>

Bibliografía.

The OM Composers Book - Vol. 2 J. Bresson, C. Agon, G. Assayag (Eds.) Collection Musique/Sciences Editions Delatour France / Ircam – 2008.

The SuperCollider Book. Wilson, S., Cottle, D. and Collins, N. (Eds). Cambridge, MA: MIT Press. 2011.