

Canicas Crónicas: Un Enfoque Procesual Mediante CDP y Soundthread

David Carratalá Domínguez

Le Cabinet Hermétique

Madrid, España

info@lecabinethermetique.com

Abstract

Esta composición electroacústica explora la disolución del pulso métrico a través del procesamiento granular y de filtrado resonante aplicado a extensas pistas de percusión. La obra busca transportar al oyente hacia una experiencia de tiempo suspendido, donde las referencias rítmicas convencionales se desvaneцен progresivamente para revelar texturas sonoras de carácter atmosférico. La particularidad del proceso creativo reside en la utilización de Soundthread —UI desarrollada por Jonathan Higgins— para controlar Composers Desktop Project (CDP), un conjunto de subrutinas desarrolladas en los años 90 y orientadas a la transformación de grabaciones dentro de la tradición de la música concrète. Una vez procesados los materiales sonoros, estos son integrados y organizados en Ableton Live, donde se realiza la edición final, espacialización y mezcla.

Palabras Clave

Soundthread; Composers Desktop Project; Música Electroacústica; Pieza Procesual; Procesamiento Granular; Músique Concrente; Música Ambient.

Introducción

El Composers' Desktop Project (CDP) lleva desarrollando software para trabajar con materiales sonoros desde 1986. Trabajando de forma cooperativa y motivados por sus propias necesidades compositivas, el equipo de CDP se ha centrado en herramientas de transformación sonora precisas, detalladas y multifacéticas a lo largo de cuatro décadas. Tras un periodo en el que muchas personas diferentes contribuyeron con diseños de software y hardware para el primer sistema Atari ST, el proyecto dio un gran salto primero cuando fue portado a Microsoft DOS en 1995 y, más tarde, cuando Trevor Wishart comenzó a centrar sus esfuerzos en programar numerosas ideas acerca de la transformación del sonido (Wishart, 1994). A día de hoy continúa siendo una de las herramientas más potentes para compositores interesados en realizar diseño sonoro o síntesis y, a pesar de ser contenido freeware, atesora funcionalidades no vistas en ninguna otra utilidad de la misma clase. El lenguaje de programación de alto nivel es C, mientras que el principal paquete de software proporcionado para la síntesis de sonido digital en el sistema es CMusic (Orton, Malham, 1987).

CMusic funciona de una manera similar a la idea de conectar generadores y modificadores de sonido en un estudio analógico. Sin embargo, en este caso, los generadores y modificadores son subrutinas codificadas

que se combinan a partir de parámetros proporcionados por el usuario:

```
C:\CDP\CDPprojects>inputfile.wav outputfile.wav  
pvalue1 pvalue2 ...
```

donde los campos pvalue corresponden a los distintos valores de configuración del efecto deseado. CDP también puede trabajar con archivos con información espectral y manipularlos.

A lo largo de los años distintas desarrolladoras han publicado front-ends, GUI o interfaces que hacen los procesos de CDP más accesibles y permiten a las compositoras interactuar con nuevos builds, siempre en modo ‘off-line’, dado que los comandos han de ser compilados antes de tener acceso al resultado de las transformaciones. Algunos de los más usados son SoundLoom, Soundshaper, ReNoise, ProcessPack, GrainMill (para síntesis granular), Shimmer (filtrado en cadena) o Brkedit (para automatizaciones). También existe el lenguaje de programación algorítmica avanzado Tabula Vigilans (TV), con mucho potencial para interactuar via MIDI con samplers o programas de partituras como MuseScore y que permite llamadas a CDP.

El UI elegido para esta composición es SoundThread (actualmente en fase beta), una interfaz de usuario multiplataforma desarrollado por Jonathan Higgins, que se ejecuta en Windows, MacOS y Linux y es compatible con más de 100 procesos CDP (Higgins, 2023). Está basada en nodos, con procesos conectados en serie o en paralelo y que luego se ejecutan como un «hilos» (thread). Cada módulo del proceso CDP muestra los parámetros relevantes y también admite la automatización. Los hilos se pueden guardar y ampliar cargando la salida como entrada.

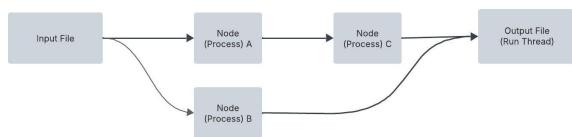


Figura 1. Esquema básico de la UI de Soundthread

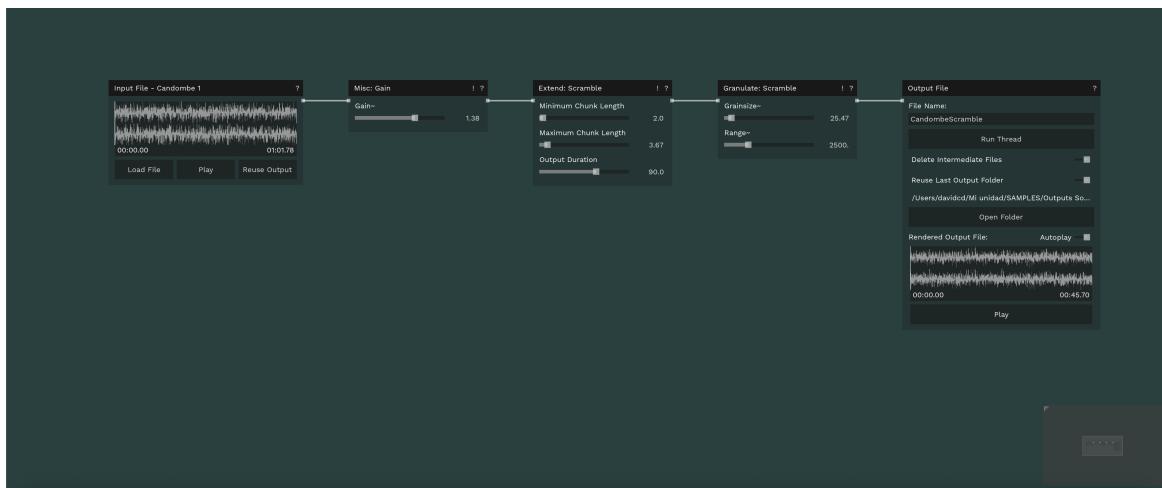


Figura 2. UI de Soundthread y nodos del primer proceso de la composición.

Piezas Procesuales

Las piezas procesuales ganaron popularidad en la música occidental a mediados del siglo XX y fueron clave en la creación de géneros posteriores como el minimalismo o la música *ambient*. Se trata de piezas musicales en las que la compositora diseña un sistema, denominado proceso, que estructura o transforma los sonidos. La pieza en sí se crea aplicando dicho proceso a cierto material sonoro primario. El proceso puede implicar aleatoriedad y azar, pero no es estrictamente necesario. Del mismo modo, puede implicar el uso de medios electrónicos o ser puramente acústico. Aunque la compositora haya diseñado el sistema y el material al que se aplica, no sabe exactamente cómo va a sonar la pieza resultante hasta que el proceso se haya ejecutado. Un hito de las piezas procesuales es ‘I Am Sitting In a Room’, de Alvin Lucier (1970), en la que el autor somete su voz hablada a constantes grabaciones y emisiones en una misma sala cerrada hasta que el discurso queda enmascarado por las reflexiones y resonancias del propio espacio (Horta Sellarès, 2023). Posteriormente, la Computer Music del s. XX no hizo más que potenciar y complejizar las ideas procesuales que ya existían en la música analógica. La precisión y velocidad del ordenador permitió aplicar procesos matemáticos, generativos y de transformación espectral que eran impensables con tecnología de cinta. Algunos ejemplos paradigmáticos de obras realizadas con esta tecnología son *Stochastic Quartet* (1962) de Iannis Xenakis o *HPSCHD* de John Cage y Lejaren Hiller (Husarik, 1969), cuyos procesos se basan respectivamente en la probabilidad estocástica y la computación algorítmica. Un ejemplo de pieza procesual que utiliza la transformación espectral sería *Mortuos Plango, Vivos Voco* (Harvey, 1981), que hace uso de todo el poder de la síntesis espectral de las nuevas computadoras de la época.

A su vez, el productor de música electrónica de baile vanguardista Richard D. James, alias Aphex Twin, quizás sea el artista popular más conocido por usar CDP en el diseño sonoro de sus composiciones. James hizo

amplio uso de esta herramienta en su EP ‘Come To Daddy’ (Warp, 1997), así como en el single ‘Windowlicker’ (Warp, 1999).

Proceso Compositivo

El objetivo de la presente pieza es explorar, de manera procesual, la disolución de los transitorios y el pulso métrico a través del procesamiento espectral, granular y de filtrado resonante aplicado a una grabación o grabaciones preexistentes. La obra busca transportar al oyente hacia una experiencia de tiempo suspendido, donde las referencias rítmicas convencionales se desvanezcan para revelar texturas sonoras de carácter atmosférico. En Soundthread, podemos acudir a varios procesos a tal efecto. El CDP cuenta con procesos que actúan tanto en el dominio del tiempo (Time Domain) como en el de la frecuencia (Frequency Domain). Particularmente nos interesan, por un lado, todos aquellos procesos que extiendan la duración de la grabación principal, así como que difuminen su espectro o que conviertan fuentes percusivas en sonidos con una estructura más armónica o inarmónica.

Para la primera de las composiciones, se ha partido de tres grabaciones de percusión medianamente largas, sin ningún tipo de acompañamiento armónico ni melódico y que hemos introducido en Soundthread como input files. Las pistas son: una grabación de una comparsa de candombe a su paso por el barrio de San Telmo en Buenos Aires (*Candombe 1.wav*), un kit de batería tocado en un compás ternario (*DrumTrackOG.wav*), el repiqueo de unas campanillas tocadas con las manos y con baqueta (*Chimes Ringing SND0729.wav*), y un disco de metal cayendo y rebotando en el suelo (*Metal Movement Disk Spinning And Falling On Floor.wav*). Las tres primeras pistas han sido sometidas a un proceso muy similar, dividido en dos partes, en Soundthread. El primero de ellos (Figura 2), consiste en procesar el Input File mediante, primero, una etapa de ganancia (*Misc: Gain*) y después los hilos *Scramble* de la carpeta *Extend* y *Granulate*. La carpeta *Extend*, según la documentación de Soundthread, contiene procesos que



Figura 3. Segundo proceso: banco de filtros afinados con ganancias automatizadas

“ajustan la longitud del sonido mediante alguna forma de repetición, incluyendo algo de aleatoriedad”. Esta aleatoriedad es importante ya que se encuentra presente en muchos procesos de CDP, por lo que replicar los resultados mostrados en la composición exactamente se puede volver una tarea difícil. La carpeta *Granulate* contiene procesos que dividen los sonidos en pequeñas muestras (o granos) y los procesan individualmente.

El proceso *Extend: Scramble* corta segmentos aleatorios de la duración total del archivo y los une de extremo a extremo mientras aleatoriza la longitud de cada fragmento para que esté entre el mínimo y el máximo. Con fragmentos máximos pequeños, esto producirá un

efecto similar a la síntesis granular, mientras que con fragmentos grandes creará nuevas frases en el material. Posteriormente, *Granulate: Scramble* recorre el archivo de principio a fin y reordena los granos con el intervalo de tiempo establecido (*Range*). Los valores de *Range* pequeños conservarán parte de la forma original del sonido, mientras que los valores grandes darán como resultado un sonido más confuso y aleatorio. Para las tres pistas de percusión, hemos escogido valores de fragmento máximo pequeños en el primer proceso y un valor de *GrainSize* de unas pocas decenas de ms. así como un *Range* de 2500 en el segundo proceso (consultar los resultados en la carpeta *Scramble* del Anexo).

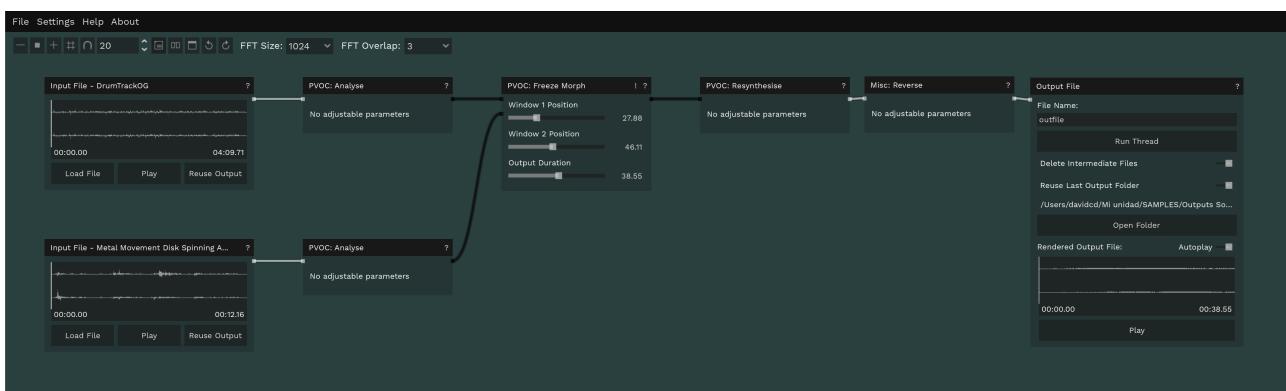


Figura 4. Tercer proceso: drone con Freeze Morph

El segundo proceso (Figura 3) se compone de un solo hilo replicado hasta cuatro veces, en el cual se hace uso del proceso *Filter: Pitched Intervals*, más una simple ganancia automatizada. Los ‘sliders’ que van acompañados de una virgulilla (~) en Soundthread pueden ser automatizados mediante el botón derecho, que abrirá la ventana de automatización. Dicha automatización tendrá lugar a lo largo de la duración del proceso, sea cual sea ésta y para ello se han dibujado diversas líneas de volumen que se van entrecruzando. El proceso *Filter: Pitched Intervals* toma una serie de filtros y los espacia equitativamente en semitonos entre las bandas más bajas y más altas. La dispersión (*Scatter*) aplica cierta aleatoriedad a estos valores, lo que hace que el sonido pase de ser muy armónico a ser más inarmónico o similar al de una campana –se ha optado por dejar este valor en el mínimo–. La ganancia de compensación (*Makeup Gain*) permite añadir un refuerzo a la señal filtrada, y el factor Q se puede automatizar, aunque hacerlo a veces produce resultados caprichosos. Este proceso producirá un sonido más consonante, ya que las frecuencias se alinearán en intervalos armónicos uniformes. De hecho, se han elegido los intervalos para que sean lo más consonantes posible mediante el ajuste del parámetro *Lowest Band*, en cada uno de los filtros resonantes. Los valores elegidos para el paso del primer archivo han sido: 164.8 Hz (E3), 220 Hz (A3), 277.1 Hz (C#4), 659.2 Hz (E5), y variaciones del acorde de La menor para el resto, a veces con la séptima y otras con la sexta. Los resultados de este segundo proceso arrojan sonidos resonantes sin un pulso fijo, similares a los que se obtendrían mediante síntesis de Karplus-Strong. Los archivos de salida se pueden escuchar en el Anexo (carpeta *Filter Pitched Intervals*).

Por último, se ha explorado otro hilo de procesos más en Soundthread para esta composición. Para crear un ‘drone’, hemos escogido dos de los files originales (*DrumTrackOG.wav* y *Metal Movement Disk Spinning And Falling On Floor.wav*) y los hemos insertado en el hilo que se muestra en la Figura 4, usando el nodo *Input File*, que nos permite trabajar con más de un sonido de partida para procesarlo en paralelo o en serie junto a otros. Ambos archivos pasan por el nodo *PVOC: Analyse*, que ejecuta una Transformada de Fourier para convertir el sonido desde el dominio del tiempo al de la frecuencia. Los parámetros de dicha transformada pueden ser modificados desde la propia UI del programa (*FFT Size* y *Overlap*). A continuación, ambas

transformadas llegan a los inputs de *PVOC: Freeze Morph*. Este proceso permite extraer ventanas de cada input y luego crear información espectral que se transforma lentamente desde las ventanas del primer archivo al segundo. Se trata de un proceso excelente para crear ‘drones’ que evolucionan lentamente con contenido espectral. Ya solo resta re-sintetizar los archivos para devolverlos al dominio temporal, y en este caso se ha añadido también un último proceso (*Reverse*) que simplemente reproducirá al revés el sonido. El resultado puede consultarse en la carpeta ‘Drone’ (*Drone Freeze Morph.wav*).

Para finalizar la pieza, los materiales resultantes han sido integrados y organizados en Ableton Live, donde se han realizado tanto la edición final, como la espacialización y la mezcla. La automatización de volumen y panning L/R han sido grabados en directo con un controlador MIDI, y tras eso se ha añadido algo de ecualización, compresión y reverberación. El resultado final de la composición se puede escuchar en la carpeta ‘Composición Final’, bajo el título ‘*Canicas Cronicas - Version Final.wav*’.

Conclusiones

El proceso compositivo descrito en el presente trabajo demuestra la vigencia y potencia de la tradición de la música concreta cuando se articula a través de herramientas contemporáneas como Soundthread, e históricas como el Composers Desktop Project (CDP). Se confirma que el enfoque procesual, heredero de pioneros como Cage, Lucier y Wishart, sigue siendo un paradigma fértil para la creación electroacústica. La pieza resultante valida un método compositivo en el que el foco creativo se desplaza desde la escritura de eventos discretos hacia el diseño de ecosistemas de transformación sonora, donde la agencia reside en la configuración y orquestación de procesos algorítmicos *off-line*. Finalmente, se deja al juicio y la percepción del/la oyente si la pieza cumple su objetivo estético inicial —la disolución del pulso métrico hacia un tiempo sonoro suspendido—. En definitiva, la presente pieza se erige como un testimonio de cómo las herramientas digitales pueden actualizar y potenciar las preocupaciones estéticas históricas de la electroacústica. El trabajo futuro podría profundizar en la automatización avanzada dentro de Soundthread para crear sistemas compositivos aún más dinámicos y responsivos.

Referencias

Wishart, Trevor. *Audible Design*. York, UK: Orpheus the Pantomime Ltd., 1994.

Orton, Richard H., and David G. Malham. "Mainframe Computer Music on Microcomputers." Paper presented at the Audio Engineering Society Convention, preprint 2479, March 1987.

Horta Sellarès, Arnau, 2023. "I am sitting in a room. Alvin Lucier y la inversión de la fenomenología" [Tesis doctoral, Universitat Autònoma de Barcelona]. https://ddd.uab.cat/pub/tesis/2024/hdl_10803_691549/ahs1de1.pdf

Husarik, Stephen. "John Cage and LeJaren Hiller: HPSCHD, 1969." *American Music* 1, no. 2 (1983): 1–21. <https://doi.org/10.2307/3051496>

Harvey, Jonathan. "'Mortuos Plango, Vivos Voco': A Realization at IRCAM." *Computer Music Journal* 5, no. 4 (1981): 22–24. <https://doi.org/10.2307/3679502>

CDP History: <http://www.composersdesktop.com/history.html>

CDP Interfaces: <https://www.composersdesktop.com/interfaces.html>

Soundshaper: <https://www.soundshaper.net/>

ProcessPack description: <https://www.composersdesktop.com/ProcessPackDescription.html>

Jonathan Higgins: <https://jphiggins.co.uk/>

Soundthread: <https://github.com/j-p-higgins/SoundThread>

Anexo: Composición y archivos

<https://drive.google.com/drive/folders/12-s2Tqu4cepklE2MG6zBiFf78pXFpdA5?usp=sharing>