



Sistemas Informáticos

Curso 2004-2005

Analizador de un espectro musical

Andrés Jiménez Sánchez
Amaia de Miguel Morate
Javier Villellas Bernal

Dirigido por:
Prof. Jaime Sánchez Hernández
Dpto. Sistemas Informáticos y Programación

Facultad de Informática
Universidad Complutense de Madrid

ÍNDICE

RESUMEN	- 3 -
SUMMARY	- 3 -
PALABRAS CLAVE	- 4 -
1.- INTRODUCCIÓN	- 5 -
2.- BASE TEÓRICA	- 5 -
2.1.- LA FÍSICA DEL SONIDO.....	- 5 -
2.1.2.- ARMÓNICOS.....	- 6 -
2.2.- EL SONIDO EN EL OÍDO HUMANO.....	- 7 -
2.2.1.- FUNCIONAMIENTO DEL OÍDO	- 7 -
2.2.2.- LA CÓCLEA.....	- 8 -
2.2.3.- ALGUNOS PARÁMETROS Y LIMITACIONES DE LA AUDICIÓN HUMANA.....	- 12 -
2.3.- MATEMÁTICAS.....	- 13 -
2.3.1.- FOURIER.....	- 13 -
2.3.2.- LAPLACE	- 17 -
2.3.3.- EL COCLEOANÁLISIS	- 17 -
3.- IMPLEMENTACIÓN	- 19 -
3.1.- HERRAMIENTAS	- 19 -
3.1.1.- GLADE	- 19 -
3.1.2.- GTK+	- 21 -
3.2.- CÓDIGO	- 22 -
3.2.1.- FMOD	- 22 -
3.2.2.- LA CLASE REPRODUCTOR.....	- 22 -
3.2.3.- CALLBACKS	- 25 -
3.2.4.- OTROS ARCHIVOS	- 27 -
3.3.- INTERFAZ.....	- 27 -
4.- APLICACIONES	- 34 -
4.1.- APLICACIONES MUSICALES	- 34 -
4.1.1.- ALGUNOS SONIDOS DE GUITARRA	- 34 -
4.1.2.- ALGUNOS SONIDOS DE DIDGERIDOO	- 35 -
4.2.- ANÁLISIS DE SONIDOS CARDÍACOS	- 37 -
4.3.- ANÁLISIS DE SONIDOS PRODUCIDOS POR RUEDAS DENTADAS.....	- 40 -
5.- ¿POR QUÉ COCLEOMATIC?	- 43 -
5.1.- ANÁLISIS DE FRECUENCIA VS. COCLEOGRAMA	- 43 -
5.2.- VENTAJAS	- 44 -
5.2.1.- PRECISIÓN.....	- 44 -
5.2.2.- PARAMETRIZACIÓN.....	- 45 -
5.2.3.- ANÁLISIS PARCIAL VS. ZOOM	- 46 -
5.3.- COMPARANDO COCLEOMATIC	- 46 -
6.- CONCLUSIONES.....	- 52 -
BIBLIOGRAFÍA	- 54 -
APÉNDICE 1: TABLA DE FIGURAS	- 55 -
APÉNDICE 2: AUTORIZACIÓN	- 56 -

RESUMEN

CocleoMatic es un analizador de espectro de sonidos cuyo objetivo es implementar una simulación realista de la percepción del sonido por parte del oído humano. En concreto simula el funcionamiento de la cóclea mediante un algoritmo desarrollado a partir de la transformada discreta de Fourier (DFT). A partir de dicho algoritmo se puede calcular el espectro de una forma de onda en términos de un conjunto de sinusoides relacionadas armónicamente, cada una con una amplitud y fase propias.

Hay una gran cantidad de analizadores de espectro de sonido pero lo que los diferencia de CocleoMatic es que están basados en la transformada rápida de Fourier con las limitaciones que ello supone. Además CocleoMatic ofrece la posibilidad de personalizar el análisis por medio de distintos parámetros para alcanzar un muy elevado grado de precisión que no consigue ninguna otra aplicación. Por otra parte CocleoMatic ofrece otras opciones como la posibilidad de grabar tus propios sonidos o guardar los análisis realizados.

SUMMARY

CocleoMatic is a sound spectrum analyzer whose aim is to implement a realistic simulation of human ear sound perception. To be precise it simulates the cochlear behaviour by means of an algorithm developed bearing in mind the Discrete Fourier Transform (DFT). This algorithm calculates the spectrum of a waveform in terms of a set of harmonically related sinusoids, each with their own amplitude and phase.

There is a wide range of sound spectrum analyzers, but what sets them apart from CocleoMatic, is the fact that they are based on the Fast Fourier Transform with their subsequent limitations. Moreover, CocleoMatic offers the possibility of a personalized analysis in order to achieve high precision results, which are not obtained by other applications. CocleoMatic also offers options such as recording your own sounds or saving finished analysis.

PALABRAS CLAVE

- Análisis
- Cóclea
- Cocleograma
- GTK
- Espectro
- Fourier
- Frecuencia
- Música
- Resonador
- Sonido

1.- INTRODUCCIÓN

Nuestro proyecto simula la percepción de tonos en el oído humano. Se trata de la construcción de un "cocleograma" que imita la excitación de la membrana basilar a diferentes frecuencias. El "cocleograma" es una representación en tiempo y frecuencia. En el eje x tenemos el tiempo y en el eje y la frecuencia, por lo que para cada instante y frecuencia podemos obtener el nivel de excitación de la membrana basilar mediante un código de color. Empezaremos explicando la estructura y funcionamiento del oído humano, seguido por los fundamentos matemáticos en los que se basa el análisis.

2.- BASE TEÓRICA

2.1.- LA FÍSICA DEL SONIDO

La base física del sonido y su interacción con el oído humano son fundamentales para comprender el funcionamiento y funcionalidad de cualquier analizador de espectro, y mucho más aun si entendemos Cocleomatic como un analizador de frecuencia basado en la manera en la que la cóclea percibe el sonido. No se trata de un algoritmo mecánico y realista desde el punto de vista exclusivamente físico, sino de una interpretación que simula la manera en la que un sonido llega a nuestro cerebro.

Para que se produzca un sonido, es necesario que un objeto vibre. Por ejemplo, cuando una persona habla, el sonido que emite es producido por las vibraciones de sus cuerdas vocales. Cuando tocamos un tambor, un pedazo de madera o de metal, una cuerda de violín, etc., estos cuerpos vibran.

En tiempo de Pitágoras, se comprobó que si una cuerda se sujetaba por la mitad, cada una de las dos mitades en las que quedaba dividida producía un sonido consonante¹ (agradable al oído) con el anterior, pero sin embargo más agudo. Este sencillo pero importantísimo hallazgo es el fundamento de la música tal y como la entendemos.

El movimiento que realiza un punto de la cuerda al vibrar consiste en un "sube y baja", un movimiento armónico simple, denominación del movimiento que se obtiene cuando los desplazamientos del punto son directamente proporcionales a las fuerzas causantes de este desplazamiento. La distancia que separa en cada instante de tiempo un punto de su centro de equilibrio (cuerda en reposo) se llama *elongación*.

¹ **Sonidos consonantes:** son aquellos que "suenan bien" al producirse simultáneamente, es decir, no generan tensión en el oyente. Físicamente, corresponden a sonidos cuya frecuencia es múltiplo una de otra (separados por una octava) o guardan una relación simple de proporcionalidad (como 3:2, la *quinta*, o 4:3, la *cuarta*).

La elongación máxima, es decir, la mitad de la distancia entre los dos extremos, es lo que se denomina *amplitud*. El número de oscilaciones completas (dobles) que realiza por segundo es la *frecuencia*, y su unidad es el hertzio (hz). Es decir, cuando decimos que el punto oscila a 100 hz queremos señalar que realiza 100 oscilaciones completas por segundo.

La presencia de un medio elástico (como el aire) permite que el movimiento del punto transmita su vibración a otras partículas del aire hasta llegar a nuestro tímpano. Por otra parte, la vibración también se transmite por la propia cuerda (que también es un medio elástico). De hecho, la suma de todas las vibraciones de los puntos determinará en la cuerda una vibración conjunta que es la que se transmite al aire.

La transmisión de energía al aire se llama onda sonora, y al movimiento conjunto de las partículas se le conoce como movimiento ondulatorio. Se llama *foco* al lugar donde se origina la perturbación, y frente de onda al punto más distante de éste.

Resulta entonces que en un instante dado todas las moléculas del aire vibran con el mismo movimiento armónico que el foco del sonido. Pero la vibración no llega a la vez a todas las partículas, sino que tarda algún tiempo, dependiendo de la distancia a la que se encuentre cada partícula del foco en la cuerda vibrante.

Este retardo en empezar a vibrar provoca que en un instante dado moléculas diferentes puedan ocupar posiciones diferentes dentro de su mismo movimiento armónico. Dos partículas pueden por tanto encontrarse en la misma fase, o bien desfasadas.

A la distancia que hay entre una partícula y la más próxima "en fase" se le llama longitud de onda.

2.1.2.- ARMÓNICOS

Cuando un objeto vibra, como por ejemplo una cuerda del piano, produce ondas complejas, que son sumas de determinadas ondas simples.

Hasta ahora hemos supuesto que la fuente sonora, la cuerda, se comportaba como un diapasón. Es decir, emitía a través del aire una onda armónica pura que viajaba por el aire. Sin embargo, la cuerda del instrumento está sujeta por los extremos. La longitud de la cuerda, así sujeta, determina la frecuencia con la que puede vibrar.

Imaginemos que oímos una flauta y después un clarinete emitiendo la misma nota. Se pueden distinguir fácilmente ambos sonidos. La razón es que aunque emiten la misma frecuencia fundamental o primer armónico, se emiten además otras frecuencias secundarias que se unen a la primera con diferentes intensidades.

Las frecuencias (incluida la fundamental) que emite a la vez el instrumento se llaman armónicos. Las distintas intensidades con las que cada instrumento emite estos armónicos forman, en conjunto, el sonido completo que oímos. Esta cualidad se conoce como timbre del instrumento. La frecuencia de los armónicos, siempre es un múltiplo de la frecuencia más baja, la fundamental. A medida que las frecuencias son más altas, los segmentos en vibración son más cortos y los tonos musicales están más próximos los unos de los otros.

Son por tanto, los armónicos los que determinan la sonoridad y timbre tanto de los instrumentos musicales como de cada uno de los sonidos que percibe nuestro oído.

2.2.- EL SONIDO EN EL OÍDO HUMANO

La percepción de un sonido por parte de nuestra capacidad auditiva está influenciada por muchos factores, tanto ajenos al sujeto que lo recibe, como inherentes a él.

La altura de un sonido es la percepción que nuestro cerebro tiene de la frecuencia. Esto nos permite clasificar algunos sonidos como agudos y otros como graves. La altura corresponde a la medida física de frecuencia. Cuanta más alta sea la frecuencia de un sonido, más agudo lo percibiremos. Generalmente, las mujeres tienen la voz más aguda que los hombres (esto es, sus cuerdas vocales vibran más rápido). En lenguaje musical se dice que un sonido agudo tiene un tono alto y que uno grave tiene un tono bajo.

Las notas musicales se caracterizan por su altura o frecuencia. En un piano, por ejemplo, a cada tecla le corresponde un sonido diferente de frecuencia. Las teclas que se hallan a la izquierda del pianista corresponden a las notas de frecuencia baja (sonidos graves), y las de la derecha son las notas de frecuencia elevada (sonidos agudos). La voz humana comprende normalmente unas dos octavas.

Cuando se introduce energía en una cuerda al pulsarla, la energía se reparte entre varios de estos modos naturales de oscilación de la cuerda. La cuerda vibrará en una superposición de todos ellos, sin vibrar en una frecuencia pura. Como ya hemos dicho anteriormente, esta combinación de frecuencias (múltiplos de la fundamental) caracteriza el sonido.

El espectro (timbre) de un instrumento tocando "re" es la traslación del mismo espectro tocando "do". Este es un caso especial de traslación de la altura.

2.2.1.- FUNCIONAMIENTO DEL OÍDO

El nivel de intensidad sonora es una medida objetiva de la intensidad del sonido, pero está lejos de representar con precisión lo que realmente se percibe. Esto se debe a que la sensibilidad del oído depende fuertemente de la frecuencia.

En general, hace falta menos intensidad para oír un sonido agudo que uno grave. Mientras que un sonido de 1.000 Hz y 0 dB ya es audible, es necesario llegar a los 50 dB para poder escuchar un tono de 50 Hz, aunque sólo un 1 por ciento de las personas pueden oír esta frecuencia a tan bajo volumen.

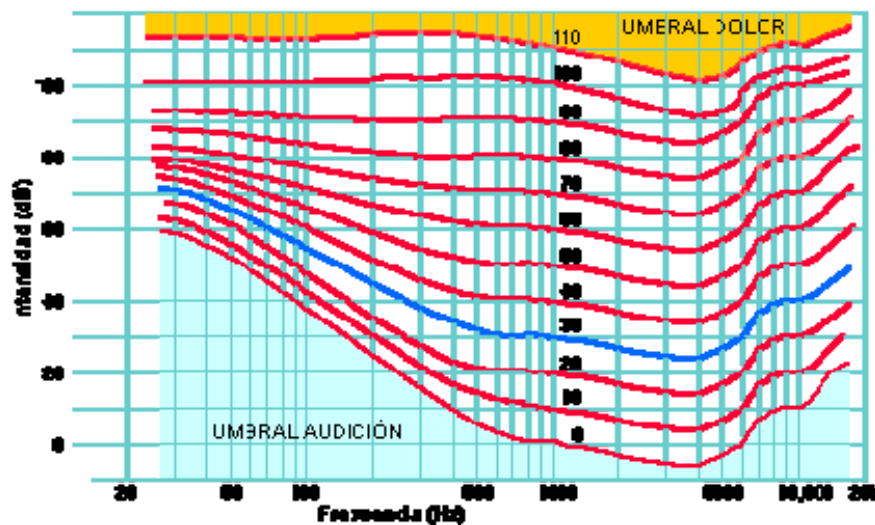


Figura 1: Gráfica de Fletcher y Munson

Nuestro oído es más sensible a unas frecuencias que a otras. La línea que marca el umbral de audición recoge los datos de los que tienen un oído muy fino. El umbral de audición de la mayoría de las personas sigue la línea azul. La línea que marca el umbral de dolor varía poco, salvo alrededor de los 4 kHz, que es la zona en donde el oído humano se muestra más sensible.

2.2.2.- LA CÓCLEA

El proceso de adquisición de información, almacenamiento, generación de memoria, evocación, y transformación del fenómeno físico en información comienza con la llegada de las ondas sonoras como una información que debe ser "entendida" por el sistema nervioso. La parte de nuestro oído encargada de esta función es la cóclea, en donde tiene lugar este proceso de traducción o transducción desde el idioma físico de una onda sonora a las señales electrofisiológicas procesadas por el sistema nervioso.

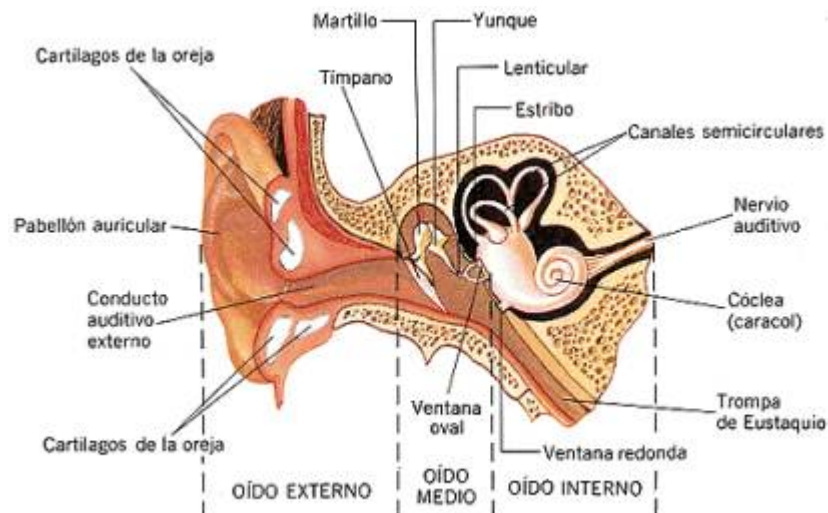


Figura 2: El oído humano

Siendo más concretos, el milagro de la audición depende de un relativamente pequeño conjunto de células que habitan en el interior de la cóclea: las células ciliadas o *resonadores*.

Estos receptores son transductores mecano-eléctricos extremadamente sensibles. Cada célula ciliada es sensible a un limitado margen de estímulos, y deben combinarse miles de receptores para dar salida a un mensaje completo. CocleoMatic simula la manera que tienen estos resonadores de interpretar el sonido y transmitirlo al cerebro, con sus aciertos e imperfecciones, con lo que hemos de prestar especial atención a estos diminutos pero importantísimos amigos.

Situémoslos dentro de la cóclea:

La espiral ósea arrollada alrededor del modiolos está separada en 3 compartimientos: la escala vestibular, la escala media y la escala timpánica.

En la escala media, limitada por la lámina reticular, la membrana de Reissner, la estría vascular, el ligamento espiral y la prominencia espiral, se encuentra la membrana basilar y el órgano de Corti, que contiene, entre otras estructuras celulares, el motor del receptor auditivo: los *resonadores* o células ciliadas.

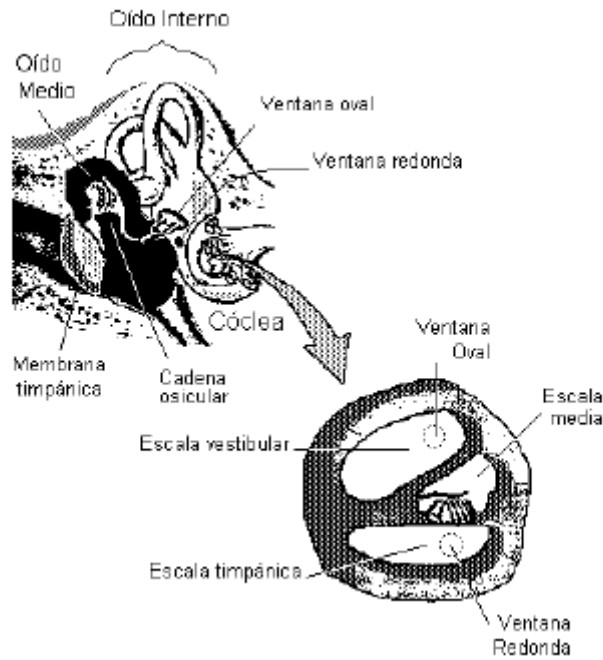


Figura 3: Esquema del oído medio e interno

La membrana basilar varía en grosor y elasticidad. Ante una frecuencia determinada se estira hasta entrar en resonancia. La parte de la membrana que entra en resonancia depende de la frecuencia. Así, podemos separar las distintas frecuencias contenidas en una única onda. Por eso podemos distinguir distintos instrumentos que tocan a la vez, así como los distintos armónicos de cada uno.

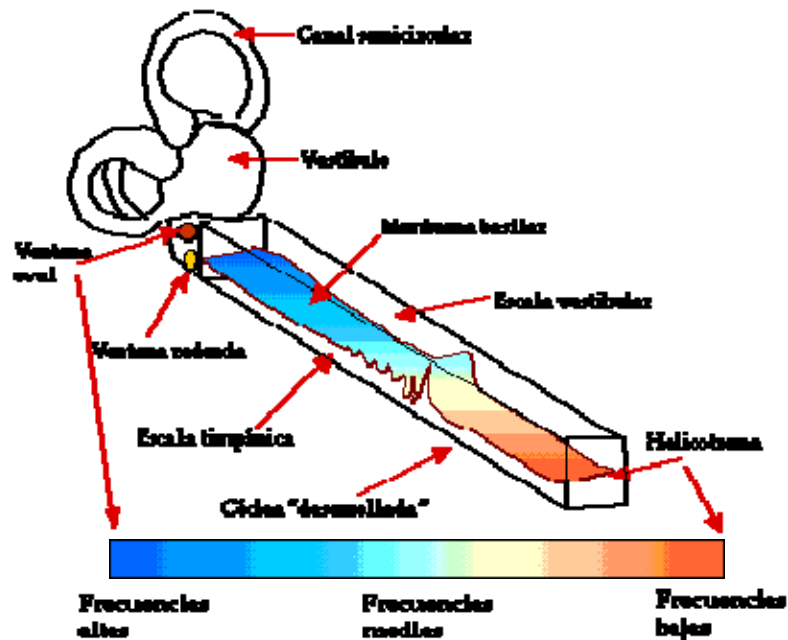


Figura 4: Cóclea desenrollada

Los resonadores, como hemos dicho, se ubican en el órgano de Corti, que se asienta sobre la membrana basilar, siguiendo la estructura en espiral

a lo largo de la misma. Son aproximadamente unas 15.000 células ciliadas para el ser humano, que se interdigitan de forma muy organizada con células de soporte. Estos receptores son sensibles a sonidos dentro de una gama de frecuencias entre unas decenas de Hertzios (Hz) hasta unos 20 kHz en el humano joven. Los sonidos naturales integrados por muchas frecuencias se descomponen en cada una de ellas, que a su vez excitan la porción que les corresponde de células ciliadas ubicadas en la membrana basilar.

La cóclea esta organizada tonotópicamente, es decir, los tonos altos producen la deformación mayor en el extremo más cercano, mientras que los tonos bajos producen la deformación más grande en el extremo más alejado.

Las células ciliadas son capaces de percibir las vibraciones a través de sus haces de estereocilias mecano-sensibles y de retroalimentar el sistema generando fuerzas mecánicas que mejoran tanto la sensibilidad como la selectividad de las frecuencias. Estas fuerzas mecánicas resultan de la motilidad (contracción o expansión) de las células ciliadas externas en respuesta al sonido, y son transmitidas a las terminaciones nerviosas a las que cada una de ellas está conectada.

Cada resonador, como hemos dicho, es sensible a una frecuencia, y alcanza el nivel de máxima excitación en un determinado instante de tiempo, siendo ésta la característica que se ha tenido en cuenta en CocleoMatic para establecer una pauta de diferenciación entre la cóclea de un individuo y la de otro. Un mismo sonido no es percibido por nuestro cerebro exactamente igual que como lo hace el de otra persona.

Cuando un sonido débil es tapado por un sonido más fuerte, decimos que se produce *enmascaramiento*. Un sonido grave e intenso puede enmascarar un sonido débil y agudo. Esto sucede porque el punto de máxima excitación de la membrana basilar para tonos de baja frecuencia está en el extremo apical de la cóclea, y para tonos de alta frecuencia, en el extremo basal.

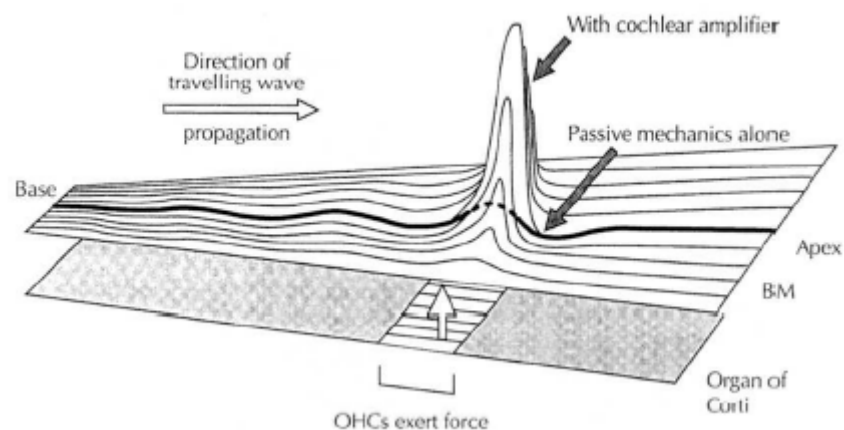


Figura 5: Fenómeno del enmascaramiento

La onda excitada por un tono agudo nunca alcanzará el punto de resonancia de un tono grave, situación que no se da en el sentido contrario. De esta manera, el oído humano no llega a distinguir y diferenciar frecuencias que se encuentran dentro del rango de audición, y que sí que se producen realmente en un sonido.

2.2.3.- ALGUNOS PARÁMETROS Y LIMITACIONES DE LA AUDICIÓN HUMANA

- *Rango de frecuencias:*

20Hz a 20KHz, lo que musicalmente abarca 10 octavas (un piano tiene del orden de 7), aunque hay personas capaces de distinguir sonidos con frecuencias superiores a los 20KHz. Componentes de frecuencia muy por encima de 20KHz pueden tener efectos perceptibles, aunque no sean apreciables como componentes del sonido. Fuera de ese rango, no hay receptores en la membrana basilar que diferencien esas frecuencias.

Comparando nuestro sistema auditivo con el de otros animales, el hombre no sale del todo mal parado, aunque no se encuentra en condiciones para presumir de contar con un oído especialmente fino. Una tortuga distingue sonidos entre 20Hz y 1KHz, un gato entre 30Hz y 50KHz, un murciélago entre 3KHz y 120KHz y un delfín entre 1KHz y 130KHz.

- *Discriminación de frecuencias:*

Podemos percibir cambios entre frecuencias con ratios aproximadas de 1.002 o 1.003 entre 1KHz y 8KHz, lo que supone a grandes rasgos 200 o 350 pasos por octava. Fuera de este rango la discriminación es más pobre, pero para la mayor parte del rango de frecuencias audibles percibimos cambios en ratios menores de 1.01, lo que da más de 60 pasos por octava, o algo menos de 1/5 de medio paso.

- *Bandas críticas:*

Nuestra audición se ve afectada por "bandas críticas" de frecuencias. El ancho de estas bandas es aproximadamente 1/3 de octava, pero varía de acuerdo con la frecuencia central. La discriminación de frecuencia para señales de una única longitud de onda es aproximadamente el ancho de una banda crítica. Un sonido se percibe con más intensidad si su energía se expande a lo largo de varias bandas críticas, en lugar de concentrarse en unas pocas. Podemos afirmar que se necesitan aproximadamente unas 30 bandas críticas para cubrir las 10 octavas de la percepción humana, produciendo 30 bandas disjuntas.

- *Algunos niveles de ruido:*

Midiendo en decibelios, distintos niveles de ruido para determinados ejemplos pueden ser:

Fenómeno Físico	Decibelios
Susurro de hojas	10 dB
Ruido en un estudio de grabación	20 dB
Ruido en una habitación tranquila	30 dB
Conversación normal	30-70 dB
Ruido en una calle tranquila	40 dB
Música a un volumen normal	50 dB
Conversación en una fiesta	60 dB
Calle ruidosa	70-80 dB
Orquesta sinfónica	90 dB
Perforadora a 2 metros de distancia	100dB
Motor de un avión a 10 metros	120 dB

Existe una medida de ruido especial, llamada sone, que está escalada a diferentes frecuencias para nuestra sensibilidad auditiva.

- *Rango de ruido:*

Desde frecuencias de 500Hz a 2000Hz, el oído humano detecta sonidos partiendo de 5dB. A partir de frecuencias menores, la sensibilidad al ruido se ve reducida, necesitando por ejemplo 75dB para llegar a distinguir un sonido a 20Hz.

A frecuencias más altas, la curva va hasta -4dB a 4000Hz y 25dB a 12000Hz.

A partir de 100dB el sonido comienza a ser molesto y a 140dB incluso doloroso.

2.3.- MATEMÁTICAS

2.3.1.- FOURIER

DFT (TRANSFORMADA DISCRETA DE FOURIER)

Gracias a Fourier se sabe que formas de onda periódicas se pueden representar mediante sumas de ondas sinusoidales relacionadas armónicamente. La transformada de Fourier descompone un ciclo de una onda en sus componentes sinusoidales.

Al multiplicar dos ondas sinusoidales con igual frecuencia, se obtiene una onda con amplitud proporcional a las dos anteriores. Sin embargo, si

difieren en la frecuencia, el resultado de la multiplicación será cero. Mediante esta multiplicación, se obtienen las ondas de frecuencia de suma y resta. Si las dos ondas tienen la misma fase, el resultado de la modulación será la multiplicación directa de las ondas. Si difieren en la fase, la diferencia es proporcional al coseno del desfase. De esta manera el valor decrece siguiendo el coseno y es cero en $\frac{\pi}{2}$ radianes (donde el coseno es cero).

Por lo tanto la medición mediante senos no tiene sentido a no ser que las dos muestras tengan la misma fase. Pero podemos ayudarnos con el coseno que obtiene resultados similares, excepto en los puntos donde el cálculo mediante senos es peor. Es allí donde los resultados de medición están en fase.

Cuando la fase de la segunda muestra se encuentra entre las fases del seno y coseno, las dos mediciones obtienen un resultado parcial. Usando la siguiente identidad, para cualquier θ podemos calcular la fase y la amplitud de la componente de la segunda onda, a partir de las muestras de seno y coseno de la primera:

$$\sin(\theta)^2 + \cos(\theta)^2 = 1$$

A parte de operar con el único ciclo de seno (y coseno) tenido en cuenta hasta el momento, se continúa con la serie armónica (2x, 3x, 4x,...) hasta la mitad de la frecuencia de muestreo. Se tendrán dos puntos de muestra por cada ciclo de prueba. Cuantos más puntos obtengamos, menor será la frecuencia fundamental con la que comenzaremos y más armónicos se podrán incluir en la transformada.

Normalmente los resultados de las muestras de seno y coseno son representados por un único número imaginario, donde la componente coseno es la parte real y el seno la parte imaginaria. Esto se hace para simplificar las ecuaciones y es posible porque la relación entre el seno y el coseno sigue las mismas reglas matemáticas que los números complejos: por ejemplo, la suma de dos números complejos se realiza sumando las partes reales e imaginarias de manera separada, de la misma manera que se sumarían las componentes de seno y coseno.

Para obtener la magnitud (que es lo mismo que la amplitud, ya que estamos interesados en el valor absoluto) basta con hacer el siguiente cálculo:

$$\sqrt{\text{Re}^2 + \text{Im}^2}$$

siendo: Re la parte real e Im la parte imaginaria.

Para expresar todo esto a través de fórmulas, tenemos:

Una función periódica $x(t)$, se puede aproximar por

$$x(t) \approx \frac{a_0}{2} + \sum_n (a_n \cos(2\pi ft) + b_n \sin(2\pi ft))$$

donde a_n y b_n se pueden obtener respectivamente por:

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cos(2\pi ft) dt$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sin(2\pi ft) dt$$

con $f = \frac{n}{T}$.

Podemos usar la relación de Euler para reescribir la versión sinusoidal de las series de Fourier en términos de exponenciales:

$$x(t) = \sum_n \alpha_n e^{i2\pi ft}$$

donde $\alpha_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-i2\pi ft} dt$

Si el periodo T converge a infinito, obtenemos la transformada de Fourier:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt$$

que convierte una función de tiempo en una función de frecuencia (x(t) es representada por X(f)).

La forma discreta (digital) de la transformada (DFT) para una serie en tiempo de longitud N es:

$$X(j\Delta f) = X_j = \sum_{k=0}^{N-1} x_k e^{-i2\pi k j / N}$$

y

$$x(k\Delta t) = x_k = \frac{1}{N} \sum_{j=0}^{N-1} X_j e^{\frac{i2\pi kj}{N}}$$

con $\Delta f = \frac{1}{N} \Delta t$

Esta relación expresada con sinusoides, es la siguiente:

$$e^{\frac{-i2\pi kj}{N}} = \cos\left(\frac{2\pi kj}{N}\right) - i \sin\left(\frac{2\pi kj}{N}\right) \quad \text{con lo que nos queda:}$$

$$X(j\Delta f) = X_j = \sum_{k=0}^{N-1} x_k \left(\cos\left(\frac{2\pi kj}{N}\right) - i \sin\left(\frac{2\pi kj}{N}\right) \right) \quad \text{Ecuación 1}$$

Estas ecuaciones permiten obtener los valores X o x de j e i respectivamente, que están fuera del rango entre 0 y $N-1$. Estos valores corresponden a las réplicas periódicas de X y x que están implícitas en la formulación discreta.

Si hacemos la sustitución:

$$z = e^{\frac{-i2\pi j}{N}}$$

entonces la ecuación de la DFT se convierte en:

$$X(j\Delta f) = X_j = \sum_{k=0}^{N-1} x_k e^{\frac{-i2\pi kj}{N}} = \sum_{k=0}^{N-1} x_k z^k = X(z)$$

Esta relación es la llamada transformada Z, que es una generalización de la DFT.

FFT – FAST FOURIER TRANSFORM

Se trata de una optimización en el tiempo de cálculo de la DFT. Por ejemplo se puede ver que las muestras de seno son cero al principio y a la mitad, por lo que no hace falta operar con ellos. Las muestras de seno pares son cero cada cuatro incrementos, cada cuarta muestra es cero cada ocho incrementos,...etc. Por lo tanto la transformada rápida de Fourier funciona exigiendo un tamaño que sea potencia de dos y opera dividiendo el proceso en grupos de dos.

2.3.2.- LAPLACE

En realidad la transformada de Fourier no es más que una particularización de la transformada de Laplace, cuya ecuación se muestra a continuación.

$$L(\omega) = \int_0^{\infty} f(t) \cdot e^{-st} \cdot dt$$

Se puede ver fácilmente como la transformada de Fourier es un caso particular de la de Laplace en la que $s = 2\pi f$.

2.3.3.- EL COCLEOANÁLISIS

Para nuestra aplicación hemos seguido un algoritmo que implementa una transformada de la que se podría decir que está a medio camino entre la transformada Z y la transformada de Laplace. El algoritmo que implementa esta transformada es el siguiente:

```

norm[i] = 2π * frecRes/sampleRate;
∀ sample hacer
{
  i = 0;
  mientras (i < numResonadores)
  {
    resSen[i] = resSen[i]*atenuacion[i] +
               sin(norm[i]*frameInProgress) * sample;
    resCos[i] = resCos[i]*atenuacion[i] +
               cos(norm[i]*frameInProgress) * sample;
  }
}

```

Ha sido creado por el profesor director de este proyecto, Jaime Sánchez Hernández.

Este algoritmo implementa el sumatorio de términos en seno y coseno que hemos visto en la **Ecuación 1**. La variable `numResonadores` equivaldría al número de células ciliadas de la cóclea que se han mencionado con anterioridad siendo cada una excitable a una frecuencia distinta. En los vectores `resSen` y `resCos` se almacena la amplitud de la onda sinusoidal para cada frecuencia y para los términos en seno y coseno respectivamente. La variable `sample` contiene la amplitud de la muestra en cada instante mientras que `frameInProgress` nos indica el número de sample que se está muestreando en ese momento.

Por último tenemos el vector `norm` que se utiliza para normalizar y cuyas componentes valen $2\pi \cdot \text{frecRes} / \text{sampleRate}$, siendo `frecRes` la frecuencia para la que se está analizando en cada instante y `sampleRate` la frecuencia de muestreo.

El factor que hace que está transformada se acerque a la transformada `Z` es el producto inicial que se hace de `resSen[i]*atenuacion[i]`. El hecho de multiplicar por `atenuacion[i]` propicia ir reduciendo la influencia que tiene la excitación de un resonador a lo largo del tiempo. Cada componente de atenuación se calcula como $1 - \text{norm}[i] / \text{limiteResonadores}$, siendo `limiteResonadores` el máximo de la energía que pueden alcanzar los resonadores.

Este máximo de energía para los resonadores se calcula como cuatro veces la dureza de la cóclea, siendo la dureza de la cóclea el número de ciclos necesarios para que un resonador llegue a su máxima excitación. Se multiplica por 4 ya que es el máximo que integraría una onda cuadrada al multiplicarla por un seno en la misma fase.

Una vez que tenemos el valor de `resSen` y `resCos` para un resonador se calcula la energía del mismo como sigue:

```
energia[i]=sqrt(resSen[i]*resSen[i] + resCos[i]*resCos[i])
            /limiteResonadores;
```

CocleoMatic permite personalizar muchos de los parámetros que se utilizan durante el análisis:

- En primer lugar podemos elegir los **límites inferior y superior** de frecuencia que acotarán el análisis.
- También podemos elegir el **número de resonadores** que queremos utilizar.
- Podemos elegir el **tipo de escala** con el que se incrementará la frecuencia a la que se excitan resonadores contiguos. La escala puede ser **lineal** o **logarítmica**.
- También podemos personalizar la **dureza de la cóclea**, es decir elegir el número de ciclos que necesita un resonador para alcanzar su máxima energía.
- Por último y debido a la complejidad del cálculo que se realiza podemos elegir una **escala horizontal** (E_h) que nos permite no realizar el análisis para todos los samples de la muestra y en su lugar hacerlo para uno de cada E_h samples. Aunque la realidad es que no se realiza el análisis de ese sample en concreto, sino de la media de los E_h samples.

3.- IMPLEMENTACIÓN

Para la implementación de nuestra aplicación hemos utilizado básicamente dos programas: Por un lado Glade/GTK+ para la creación de la interfaz y generación del código en C, y por otro lado Microsoft Visual Studio .NET 2003.

3.1.- HERRAMIENTAS

3.1.1.- GLADE

Glade es una herramienta de desarrollo visual de aplicaciones mediante GTK/GNOME y tiene licencia GPL (General Public License).

Glade puede crear la interfaz de usuario de las aplicaciones de dos formas diferentes, o bien generando código fuente, o bien cargando dinámicamente un fichero XML de descripción de la interfaz en tiempo de ejecución. En nuestro caso, interesa ver la primera:

Al abrir un proyecto de Glade, se abren cuatro ventanas: Ventana principal, Paleta, Árbol de Widget y Propiedades:

En la ventana principal aparecen las ventanas y diálogos que le hayamos añadido a la aplicación. Además es en esta ventana en la que se da la opción de generar el código. Al pulsar el botón construir, se genera un proyecto para Microsoft Visual con los archivos callbacks.c, callbacks.h, interface.c, interface.h, support.c y support.h. De estos archivos el usuario sólo debe editar callbacks.c y callbacks.h, que contienen las funciones manejadoras de señales y de "callback".

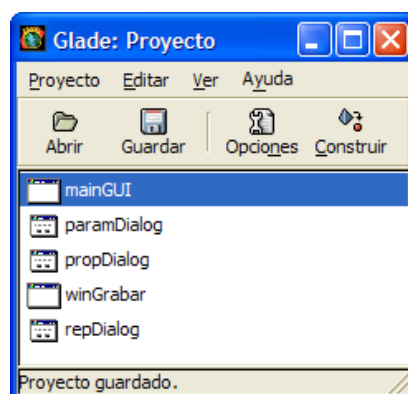


Figura 6: Aspecto de la ventana principal de un proyecto Glade

En la paleta de widgets (un widget es cualquier elemento de la interfaz, como por ejemplo una ventana, un botón o una zona de dibujo) se pueden ir seleccionando los diferentes elementos que deseemos introducir en nuestra interfaz. Basta con pinchar sobre el widget elegido y pinchar otra vez en el

lugar destino. La única restricción es que la mayoría de los widgets deben estar incluidos en un contenedor (por ejemplo una caja, un panel o un marco). En el caso de que se intente ignorar esto último, Glade nos avisará de la imposibilidad de colocar ese widget sobre algo que no sea un contenedor.

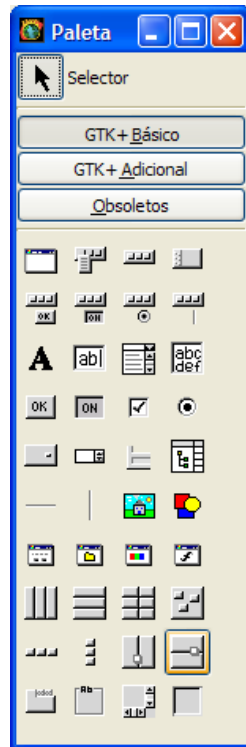


Figura 7: Aspecto de la paleta de widgets

En la ventana de árbol de widget aparece en cada momento la estructura, en forma de árbol de nuestra interfaz.

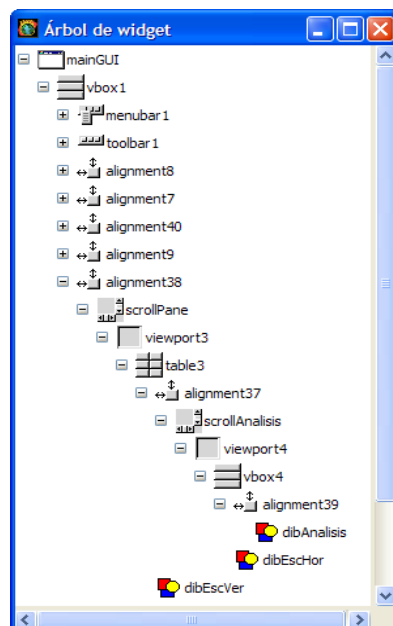
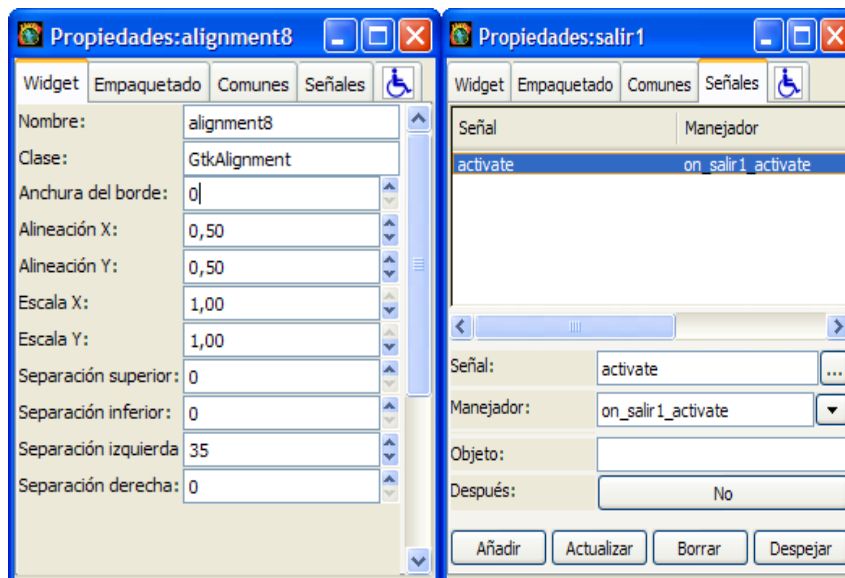


Figura 8: Aspecto de la ventana del árbol de widget

Para definir las propiedades de los diferentes widgets disponemos de la ventana de propiedades. Al pulsar sobre cualquier elemento del árbol de widgets, nos aparece la ventana de propiedades correspondiente. En ella podemos editar las diferentes propiedades de los elementos de la interfaz, dependiendo del tipo de widget del que se trate. Por mencionar alguno, se podría cambiar por ejemplo el nombre, la posición, la alineación o el tamaño. Pero lo más importante a considerar en esta ventana, es la pestaña de señales, en la que podremos definir las diferentes señales que deseemos asociarle al widget. Si queremos que un elemento de la interfaz responda a un determinado evento, deberemos crear aquí una señal adecuada. En el archivo callbacks aparecerá declarado el correspondiente manejador, que nosotros tendremos que programar para que se realice lo deseado.

**Figura 9: Aspecto de la ventana de propiedades**

3.1.2.- GTK+

GTK+ es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI), principalmente para los entornos gráficos GNOME, XFCE y ROX de sistemas Linux, aunque también es compatible con Windows. Es software libre (bajo la licencia LGPL), multiplataforma y parte importante del proyecto GNU.

Inicialmente fue creado para construir el programa gráfico GIMP (GTK+ es la abreviatura de GIMP toolkit, conjunto de rutinas para GIMP) y se ha diseñado para permitir programar con lenguajes como C, C++, Perl o Python.

GTK+ se basa en tres bibliotecas:

- GLib: biblioteca de bajo nivel, estructura básica de GTK+ y GNOME. Proporciona manejo de estructuras de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución (runtime) como ciclos, hilos, carga dinámica o un sistema de objetos.
- Pango: biblioteca para el diseño y renderizado de texto. Hace especial hincapié en la internacionalización y es el núcleo para manejar las fuentes y el texto.
- ATK: biblioteca para crear interfaces con gran accesibilidad para personas discapacitadas. Pueden usarse útiles como lupas de aumento, lectores de pantalla, o entradas de datos alternativas al clásico teclado o ratón de ordenador.

3.2.- CÓDIGO

3.2.1.- FMOD

Nuestra aplicación utiliza la clase Reproductor para todo lo relativo a la reproducción y grabación. Esta clase hace uso de FMOD, una librería multiplataforma para el manejo de sonido y música en videojuegos, demos, aplicaciones multimedia, etc. Está orientada especialmente para su uso en la programación de videojuegos. Algunas de las plataformas donde se encuentra disponible son: Windows, Linux, Windows CE, incluso está disponible para algunas consolas como por ejemplo: GameCube, PS2 y XBOX. Otra de las ventajas de esta librería es que está soportada por varios compiladores, entre ellos tenemos a Microsoft Visual C++, Watcom, Borland, LCC-WIN32, Metrowerks/CodeWarrior, GCC, MinGW & CygWin, Delphi, Kylix, Visual Basic. Algunos de los formatos soportados por esta librería son los siguientes: MOD, S3M, XM, IT y MID para música, y WAV, MP2, MP3 y OGG para sonidos. De estos últimos a nosotros nos interesa el formato WAV.

Para su utilización solamente tendremos que incluir el archivo de cabecera "fmod.h" en la clase Reproductor.

3.2.2.- LA CLASE REPRODUCTOR

La clase Reproductor tiene los siguientes atributos:

- **int canal:** Canal a utilizar para la reproducción.
- **int grabacion:** Canal a utilizar para la grabación.
- **bool grabando:** Variable que indica si se está grabando.
- **FSOUND_SAMPLE *sample:** Estructura en la que almacenamos los datos del WAV de reproducción.

- **FSOUND_SAMPLE *sampleG:** Estructura en la que almacenamos los datos del WAV de grabación.

Y los siguientes métodos:

- **Reproductor(int):**
Constructor que recibe como parámetro el indicador de canal a utilizar para la reproducción del archivo analizado
- **~Reproductor():**
Destructor de la clase.
- **void reproducir():**
Reproduce el canal de reproducción.
- **void grabar(char):**
Graba hasta 120 segundos en sampleG, en mono o estéreo, dependiendo de lo que se le pase por parámetro
- **void pararGrabacion():**
Detiene la grabación en curso.
- **bool pausar():**
Pausa el sonido de canal de reproducción.
- **bool depausar():**
Reanuda el sonido de canal de reproducción.
- **bool parar():**
Detiene el sonido del canal de reproducción.
- **void setVolumen(int):**
Establece el volumen del canal de reproducción con el valor pasado por parámetro.
- **int getVolumen():**
Devuelve el volumen del canal de reproducción.
- **bool cargarWave(const char*):**
Carga el archivo WAV pasado por parámetro en memoria.
- **int getPosicion():**
Devuelve la posición del cursor en reproducción.
- **bool estaGrabando():**
Devuelve true si se está grabando y false en otro caso.
- **int getPosicionGrabacion():**

Devuelve la posición del cursor en grabación.

- **bool estaReproduciendo():**
Devuelve true cuando se está reproduciendo el canal de reproducción y false en otro caso.
- **void guardarGrabacion(char*, int):**
Guarda la grabación en curso en el archivo pasado por parámetro.
- **bool setDriverGrabacion(int):**
Establece como driver de grabación el que se encuentra en la posición pasada por parámetro.
- **int getNumDriversGrabacion():**
Devuelve el número de drivers de grabación disponibles.
- **bool setDriverReproduccion(int):**
Establece como driver de reproducción el que se encuentra en la posición pasada por parámetro.
- **int getNumDriversReproduccion():**
Devuelve el número de drivers de reproducción disponibles.
- **void setMixer():**
Establece el mezclador.
- **const char* getDriver(int):**
Devuelve el nombre del driver de reproducción que se encuentra en la posición pasada por parámetro.
- **const char* getDriverGrabacion(int):**
Devuelve el nombre del driver de grabación que se encuentra en la posición pasada por parámetro.
- **void setMetodoReproduccion(int):**
Establece el método de reproducción: Si el parámetro vale 1, se usará el DirectSound driver, si vale 2 se usará el Windows Multimedia driver y si vale 3 no se usará driver.
- **void setEstaGrabando(bool):**
Cambia el valor del atributo grabando con el pasado por parámetro.
- **void reproducirGrabacion():**
Reproduce el canal de grabación.
- **bool pararGrabacionEnReproduccion():**

Detiene la reproducción de la grabación en curso.

- **bool estaReproduciendoGrabacion():**
Devuelve true cuando el reproductor está reproduciendo el canal de grabación y false en otro caso.
- **bool estaPausado():**
Devuelve true si se ha pausado el canal de reproducción y false en otro caso.

3.2.3.- CALLBACKS

En el archivo callbacks.c aparecen todas las funciones manejadoras de señales, para responder a los eventos creados por el usuario al pulsar sobre las distintas opciones de la aplicación. En callbacks.h, aparte de las cabeceras de las funciones, hemos declarado varias estructuras:

- La estructura Parámetros

```
struct Parametros
{
    double frecInf;
    double frecSup;
    double dureza;
    int numResonadores;
    int escalaH;
    bool logaritmica;
};
```

que almacena los parámetros del análisis: las frecuencias inferior y superior a considerar, la dureza de la cóclea, el número de resonadores, la escala y el tipo de escala (lineal o logarítmica).

- La estructura ParametrosReproductor

```
struct ParametrosReproductor
{
    int metodoRep;
    int driverRep;
    int driverGrab;
    bool mono;
};
```

que almacena los parámetros del reproductor: el método de reproducción, los drivers de reproducción y grabación, y el número de canales (mono o estéreo).

- La estructura `typGraphics`

```
typedef struct
{
    GdkDrawable *pixmap;
    GdkGC *gc;
} typGraphics;
```

que almacena un mapa de pixels y su correspondiente información de color.

A continuación detallamos las funciones más importantes incluidas en `callbacks.c` que no se corresponden con funciones manejadoras de señales:

- **`void dibujar(GtkWidget *)`:**

Esta función realiza el análisis mediante el algoritmo ya descrito y lo muestra por pantalla mediante un dibujo bidimensional que podría interpretarse de forma tridimensional. Las tres dimensiones serían la escala horizontal (tiempo), la escala vertical (frecuencia) y por último el color de cada punto que se podría interpretar como la tercera dimensión.

El color nos dice el nivel de excitación de cada resonador en cada instante de tiempo y lo calculamos a partir del resultado del algoritmo haciendo uso de la función `dameColor` para obtener el color con el que dibujar ese punto.

- **`GdkGC *dameColor(double)`:**

Esta función devuelve un puntero a un `GdkGC` (estructura que almacena información de color), a partir de una magnitud que se le pasa por parámetro y que corresponde a la energía calculada para un resonador. Para una magnitud menor o igual que 0.1, se usan distintos tonos de azul oscuro; para una magnitud mayor que 0.1 y menor o igual que 0.2, se usan distintos tonos de azul claro; para una magnitud mayor que 0.2 y menor o igual que 0.35, se usan distintos tonos de verde; para una magnitud mayor que 0.35 y menor o igual que 0.5, se usan distintos tonos de amarillo; para una magnitud mayor que 0.5 y menor o igual que 0.65, se usan distintos tonos de naranja; para una magnitud mayor que 0.65, se usan distintos tonos de rojo.

- **`void dibujarWave(GtkWidget *)`:**

Esta función dibuja la onda del archivo WAV en curso sobre la zona de dibujo correspondiente. Cuando se abre un archivo se calcula el parámetro de escala horizontal (E_h) que se utilizará

posteriormente en el análisis de modo que se ajuste el dibujo al tamaño de la zona de dibujo actual. Esto viene a significar que se dibujan E_h samples con distintas amplitudes en una misma franja vertical del dibujo.

- **void dibujarEscVer(GtkWidget *):**

Esta función dibuja la escala vertical del análisis sobre la zona de dibujo correspondiente. Dibuja la escala dividiendo el intervalo de frecuencias en 10 partes equidistantes. La escala muestra el tiempo en hercios.

- **void dibujarEscHor(GtkWidget *):**

Esta función dibuja la escala horizontal del análisis sobre la zona de dibujo correspondiente. Dibuja la escala dividiendo el intervalo de tiempo en 10 partes equidistantes que a su vez se dividen en otras 10 partes también equidistantes. La escala muestra el tiempo en milisegundos.

- **void muestraInfo(GtkWidget *, gdouble, gdouble):**

Esta función muestra la posición del cursor sobre la zona de dibujo del análisis y la fundamental para ese instante. Esta información se escribe en el panel que está inmediatamente debajo de la zona de dibujo.

3.2.4.- OTROS ARCHIVOS

Los archivos main.c, interface.c, interface.h, support.c y support.h son archivos no editables generados automáticamente por Glade y contienen información acerca de la interfaz de la aplicación.

3.3.- INTERFAZ

La interfaz de nuestra aplicación presenta el siguiente aspecto:

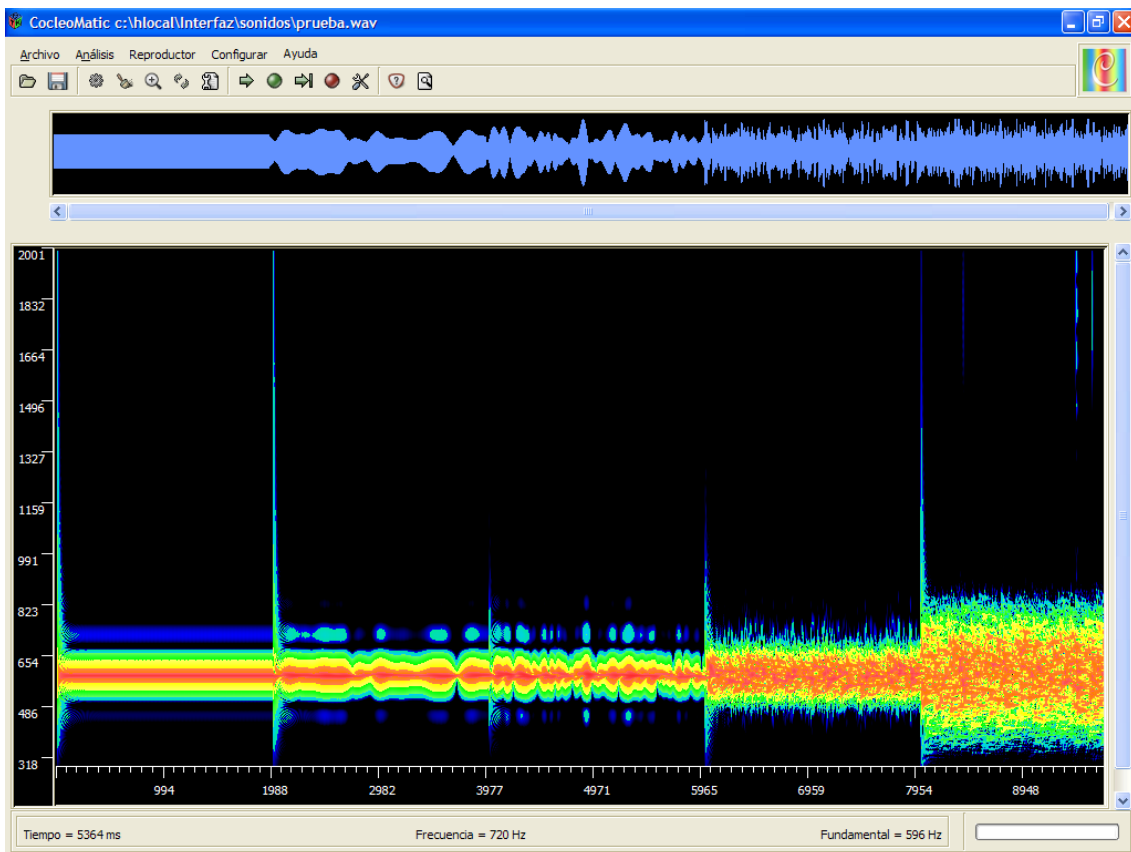


Figura 10: Aspecto de la interfaz de la aplicación

Se pueden observar dos zonas de dibujo, la superior corresponde a la onda de entrada y la inferior al análisis. Inmediatamente debajo de la zona de análisis se aprecian tres números, que van cambiando al pasar el cursor sobre el dibujo: El primero corresponde con la posición del cursor en el eje x (tiempo), el segundo con la posición en el eje y (frecuencia), y el tercero con la fundamental correspondiente a ese instante de tiempo.

A continuación se detallan las opciones del menú:

- Archivo/ Abrir:

Permite abrir un archivo .wav. En caso de intentar abrir un archivo que no tenga extensión wav, se informará al usuario del error.

Cuando se abre un archivo la aplicación calcula la escala horizontal adecuada para ajustar el dibujo de la forma de onda al tamaño de la zona de dibujo. Esta escala se podrá cambiar posteriormente mediante la opción de configurar parámetros.

- Archivo/ Guardar imagen:

Guarda una imagen en formato jpg del análisis actual. Esta opción sólo estará activa cuando haya un análisis presente.

- Archivo/ Salir:

Cierra la aplicación.

- Análisis/ Analizar:

Analiza el archivo abierto. Esta opción sólo estará activa después de haber abierto un archivo. En caso de que el archivo a analizar sea estéreo, se dará la opción de elegir el canal analizar (izquierdo o derecho).

Cuando se selecciona esta opción mientras se está realizando un análisis la aplicación detendrá la realización del mismo.

- Análisis/ Limpiar:

Limpia las zonas de dibujo de la onda de entrada y del análisis. Asimismo ya no habrá ningún archivo wav cargado.

Por otro lado libera todas las estructuras de datos utilizadas para almacenar los dibujos y parámetros del análisis.

- Análisis/ Zoom:

Realiza un zoom (análisis más detallado) sobre la zona seleccionada del análisis. En caso de estar hecho ya un zoom, se volverá al análisis original. Esta opción solamente estará activa en caso de que haya algo seleccionado o de que previamente se haya hecho zoom. La forma de seleccionar es pulsando el botón izquierdo del ratón, arrastrando sobre la zona deseada y finalmente soltando el botón. Para deshacer la selección basta con pulsar el botón derecho del ratón.

Para realizar el zoom la aplicación mantiene el mismo número de resonadores que tenía antes de hacer el zoom pero aplicado al nuevo intervalo de frecuencias. Por otro lado recalcula la escala horizontal para que sea proporcional a la que había antes pero con el nuevo intervalo de tiempo. La escala no podrá ser nunca menor que uno.

Cuando se realiza un zoom se realiza también para el dibujo de la forma de onda del sonido de entrada. La aplicación almacena los dibujos tanto del análisis como de la forma de onda previas al análisis para poder regresar del zoom de forma instantánea sin tener que volver a realizar el análisis.

- Análisis/ Cambiar canal:

Si el archivo cargado es estéreo, esta opción permite realizar el análisis del canal que no esté presente en ese momento. Asimismo aparecerá la onda de entrada correspondiente. Esta opción sólo estará activa en caso de haber abierto un archivo estéreo.

Cuando se pulsa la opción de cambiar canal la aplicación pasa a realizar el análisis del nuevo canal de forma automática. Además, como en el caso del zoom, se almacenan los dibujos tanto de la forma de onda como del análisis del canal anterior para poder volver a cambiar de canal sin necesidad de recalculr nada.

- Análisis/ Cambiar parámetros:

Al pulsar sobre esta opción del menú aparece la siguiente ventana:

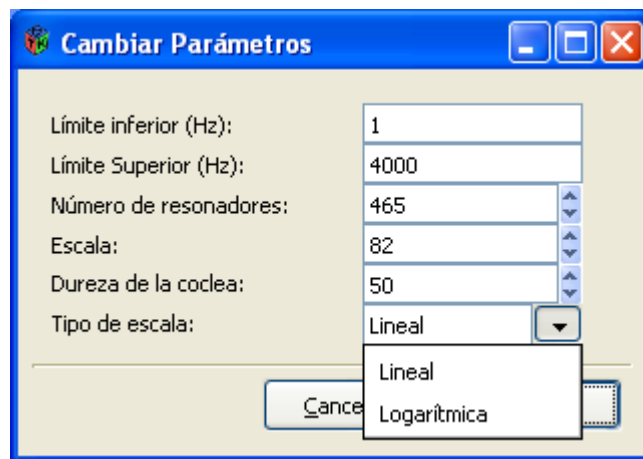


Figura 11: Diálogo Cambiar Parámetros

En ella se pueden cambiar los parámetros del análisis actual. Siempre aparecen unos parámetros por defecto que tratan de ajustar el análisis a la zona de dibujo visible. Como se puede observar en la imagen, se puede elegir entre dos tipos de escala: lineal y logarítmica. Por defecto siempre estará seleccionada la escala lineal.

Cuando se cambia el parámetro de escala horizontal se redibuja la forma de onda ajustándola a la nueva escala.

Si el usuario introdujera algún parámetro no válido la aplicación muestra un mensaje de error e ignora el valor introducido.

- Reproductor/ Play:

Permite reproducir el archivo wav cargado. En caso de haber abierto un archivo estéreo, únicamente reproduce el canal seleccionado. Durante la reproducción, en la zona de dibujo de la onda de entrada aparecerá

una línea vertical que se irá moviendo acorde con la reproducción. Esta opción solamente estará activa en caso de haber algún archivo cargado.

- Reproductor/ Stop:

Detiene la reproducción en curso, en caso de haber alguna.

- Reproductor/ Pause:

Pausa la reproducción en curso. Si se hubiera pausado con anterioridad, reanuda la reproducción.

- Reproductor/ Grabar:

Al pulsar sobre esta opción del menú aparece la siguiente ventana:



Figura 12: Ventana del CocleoGrabador

Al pulsar sobre el botón central, se inicia la grabación y al volver a pulsarlo, se detiene. Pulsando sobre el botón "Reproducir", se podrá escuchar el sonido grabado y el botón "Guardar" permitirá guardar la grabación en un archivo de sonido. En caso de cerrar la ventana sin haber guardado la grabación, se advertirá de ello al usuario.

- Reproductor/ Configuración:

Al pulsar sobre esta opción del menú aparece la siguiente ventana:



Figura 13: Ventana Configurar reproductor

En ella se pueden elegir los controladores a utilizar para la reproducción y la grabación, entre aquellos que se encuentran en el equipo desde el que se ejecuta la aplicación. Asimismo se puede seleccionar el formato de grabación (mono o estéreo).

- Configurar/ Amplificar entrada:

Esta opción amplifica la onda de entrada, de manera que la amplitud máxima sea 1. Cuando se selecciona esta opción se redibuja la forma de onda amplificada. Esta amplificación se tendrá luego en cuenta a la hora de realizar el análisis pero no a la hora de reproducir el sonido.

- Ayuda/ Acerca de:

Muestra un diálogo con información acerca de la versión de la aplicación, junto con los nombres de los creadores.

Los botones (de izquierda a derecha) permiten las siguientes opciones:

- Botón 1: Igual que Archivo/ Abrir
- Botón 2: Igual que Archivo/ Guardar imagen
- Botón 3: Igual que Análisis/ Analizar
- Botón 4: Igual que Análisis/ Limpiar
- Botón 5: Igual que Análisis/ Zoom
- Botón 6: Igual que Análisis/ Cambiar canal
- Botón 7: Igual que Análisis/ Cambiar parámetros
- Botón 8: Igual que Reproductor/ Play
- Botón 9: Igual que Reproductor/ Stop
- Botón 10: Igual que Reproductor/ Pause
- Botón 11: Igual que Reproductor/ Grabar
- Botón 12: Igual que Reproductor/ Configuración
- Botón 13: Igual que Ayuda/ Acerca de
- Botón 14: Al pulsar sobre este botón aparece una ventana con el siguiente aspecto:

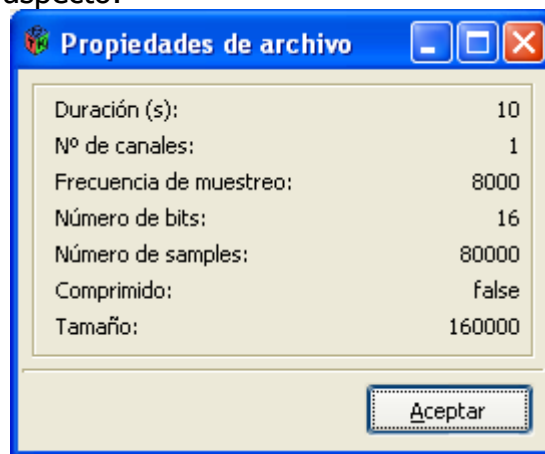


Figura 14: Ventana Propiedades de archivo

En ella aparecen las propiedades del archivo wav cargado. Esta opción solamente estará activa en caso de haber un archivo cargado.

4.- APLICACIONES

Vamos a estudiar algunos ejemplos de análisis efectuados con CocleoMatic para entender su funcionamiento.

4.1.- APLICACIONES MUSICALES

4.1.1.- ALGUNOS SONIDOS DE GUITARRA

Veamos dos ejemplos, en los que una guitarra eléctrica reproduce la misma secuencia de notas. La secuencia es C, D, F, y por último dos notas tocadas a la vez (C + G).

- En el primer ejemplo se trata de una guitarra con una ecualización totalmente limpia:

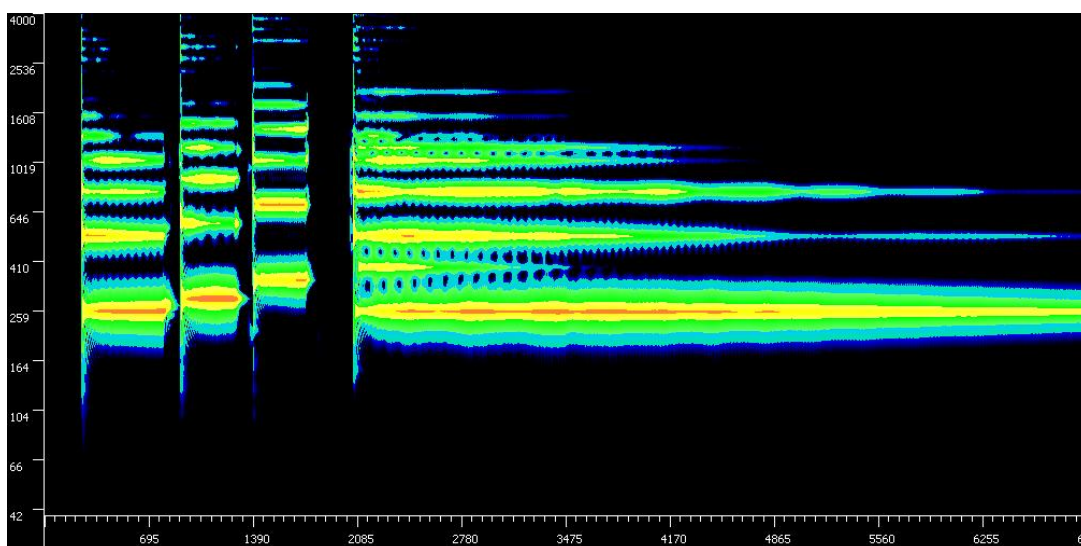


Figura 15: Análisis de sonido de guitarra 1

Se aprecian perfectamente los cambios de una nota a otra, con sus respectivas fundamentales y armónicos. Es interesante observar como se van incrementando las frecuencias y cómo en el último toque se diferencian claramente las dos notas simultáneas, el mismo C del primer toque más el G.

- En el segundo ejemplo se reproduce la misma secuencia, pero ahora se ha añadido a la guitarra distorsión. La distorsión se consigue mediante la sobrecarga de la señal de entrada a un amplificador. Es una buena oportunidad para comprobar como una misma nota varía su espectro según los timbres de los instrumentos que la reproducen.

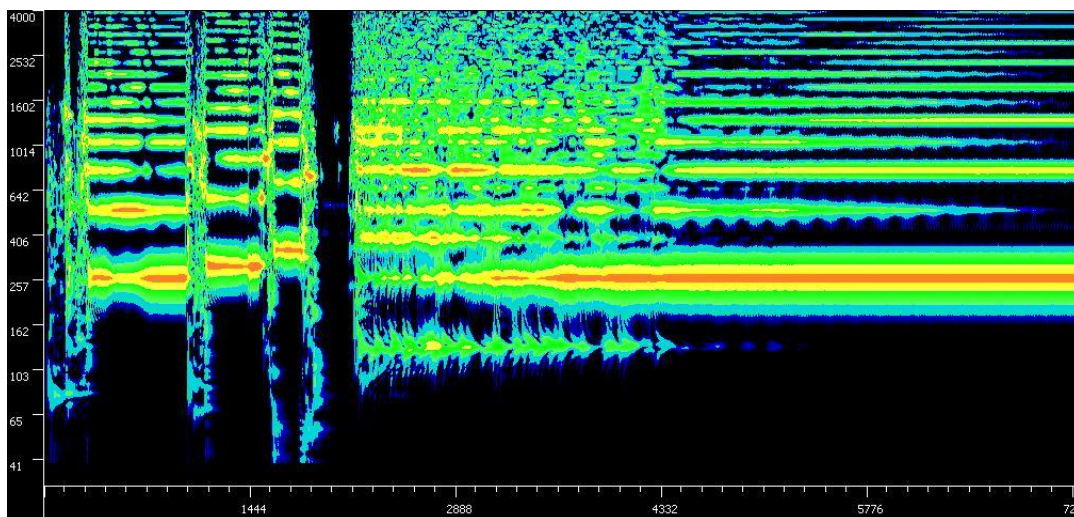


Figura 16: Análisis de sonido de guitarra 2

Observamos que se trata de las mismas fundamentales, ya que se están tocando las mismas notas, y como se ha dicho anteriormente, una misma nota tiene la misma fundamental sea quién sea quién o qué la reproduzca. La distorsión introduce más riqueza en el sonido, lo que se ve traducido en una generación de muchos más armónicos y una excitación de muchos más resonadores. El último *Do* tocado termina con un acople de la guitarra debido a la saturación, lo se traduce en casi un armónico puro que podemos distinguir claramente al final del análisis.

4.1.2.- ALGUNOS SONIDOS DE DIDGERIDOO

El didgeridoo es un instrumento de viento ancestral utilizado por los aborígenes de Australia. Se supone que tiene unos 20.000 años de existencia de acuerdo con la datación de algunas pinturas rupestres en las que aparece el instrumento, aunque los propios aborígenes le dan una antigüedad de hasta 40.000 años.

Básicamente es un tubo parecido a una gran trompeta que se hace sonar haciendo vibrar los labios en uno de sus extremos. Una de sus particularidades es que se puede tocar durante un tiempo ilimitado mediante una técnica denominada respiración circular que consiste en mantener continuamente una cierta presión de aire en la boca.

Es un instrumento que aporta una riqueza armónica brutal, cuyo análisis espectral se convierte en toda una experiencia digna de observar.

A continuación se estudian algunos ejemplos de análisis de sonidos producidos por un didgeridoo:

- Este ejemplo hace el análisis de un sonido de didgeridoo en el que se incluye mucho movimiento de lengua (una especie de pronunciación de la "doble rr" española):

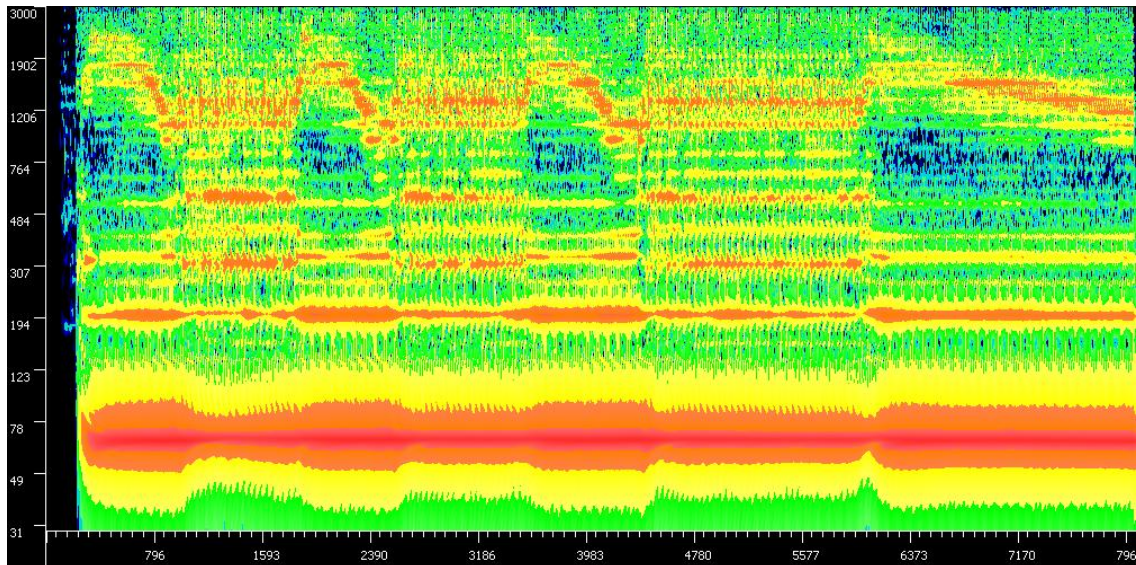


Figura 17: Análisis de sonido de didgeridoo 1

La banda roja ancha que se ve en la parte inferior del análisis se corresponde con un sonido grave que se escucha constantemente y de forma predominante.

Las "ondas" que se forman entre 1000Hz y 2000Hz aproximadamente corresponden al movimiento de la lengua. También se pueden apreciar los momentos en los que se suelta la lengua.

- En este ejemplo se hace el análisis de un sonido de didgeridoo en el que se imita el salto de un canguro:

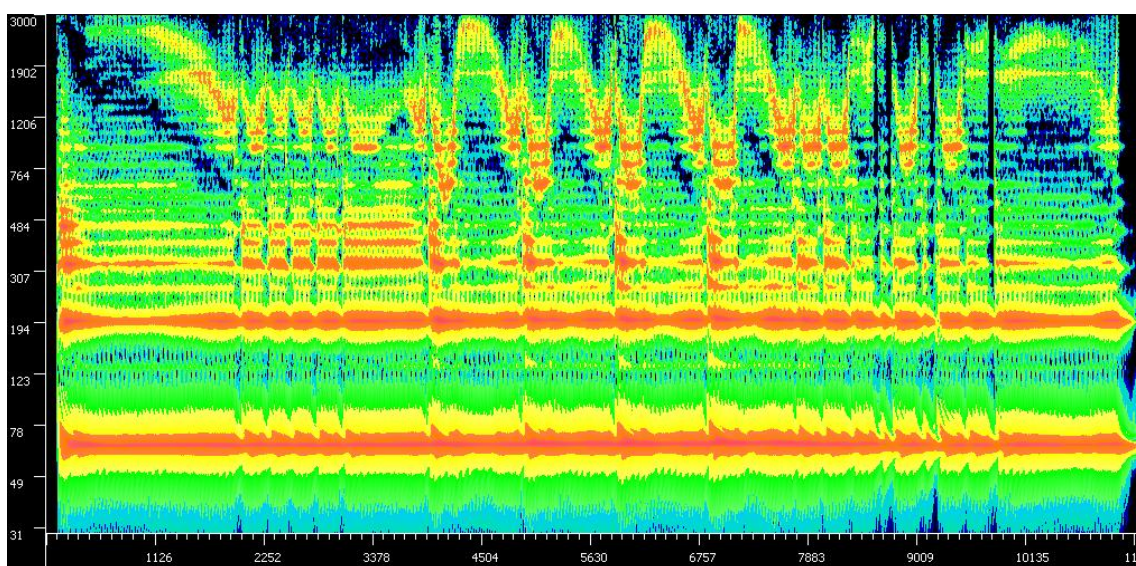


Figura 18: Análisis de sonido de didgeridoo 2

Otra vez se ve claramente la fundamental y los armónicos, y en las frecuencias superiores la simulación del salto del canguro, que se consigue mediante la introducción de u's e i's.

4.2.- ANÁLISIS DE SONIDOS CARDÍACOS

Vamos a ver la posible aplicación del cocleograma obtenido en nuestro análisis al estudio de los sonidos del corazón. Para ello vamos a empezar con una pequeña introducción sobre la estructura y funcionamiento del corazón.

A continuación vemos una imagen de un corazón.

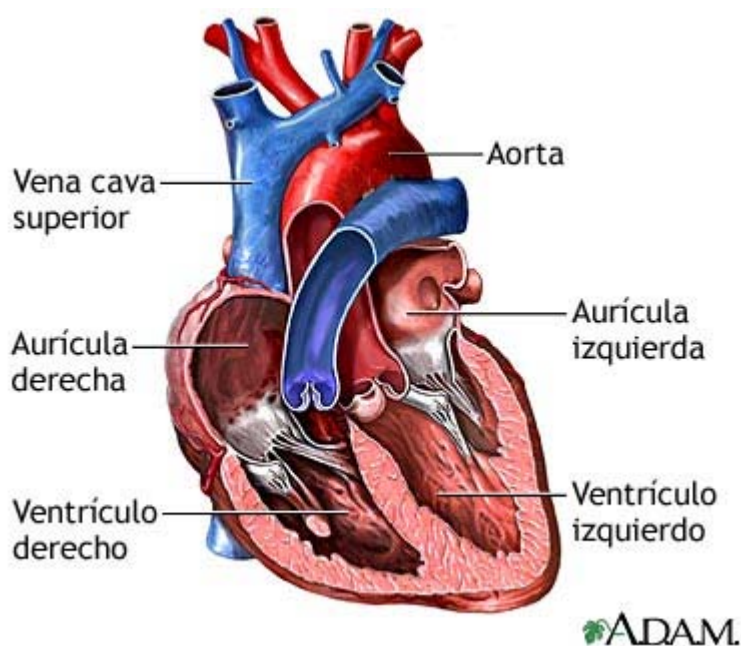


Figura 19: Las partes del corazón

El corazón contiene 4 cámaras; las dos superiores o aurículas y las dos inferiores o ventrículos. La sangre entra en la aurícula derecha y pasa a través del ventrículo derecho, el cual bombea la sangre a los pulmones donde ésta se oxigena. La sangre oxigenada es traída de nuevo al corazón por medio de las venas pulmonares que entran a la aurícula izquierda. De allí, la sangre fluye al ventrículo izquierdo, el cual la bombea hacia la aorta que distribuye sangre oxigenada a todas las partes del cuerpo.

Auscultar: método que se basa en la escucha de los distintos sonidos del cuerpo con la ayuda de un estetoscopio durante un examen físico. Los médicos escuchan el sonido de pulmones, corazón e intestinos para evaluar la frecuencia, intensidad, duración y calidad de los mismos.

Los sonidos normales del corazón se denominan S1 y S2. S1 es el latido más destacado que corresponde a la contracción de los ventrículos (sístole) y S2 el de la contracción de las aurículas (diástole). Estos sonidos se producen al cerrarse las válvulas que controlan la dirección del flujo sanguíneo. A continuación vemos el análisis de un corazón sano.

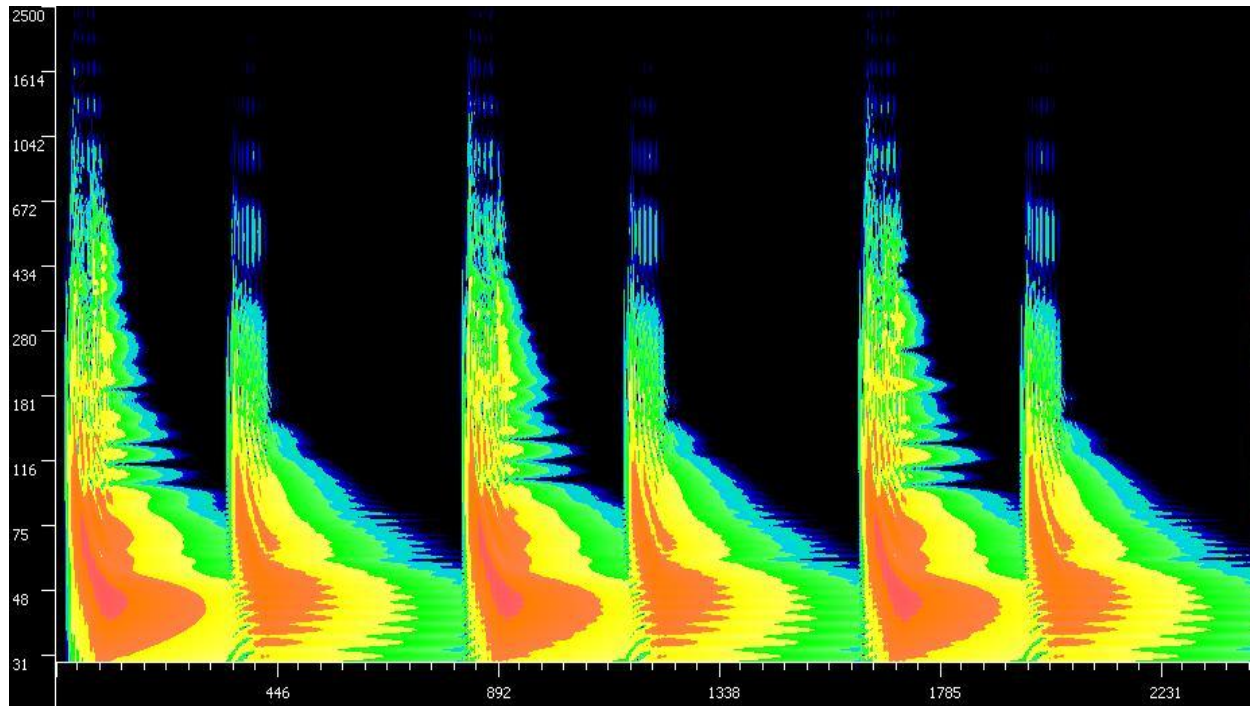


Figura 20: Análisis de sonido de un corazón sano

Como el corazón está dividido en parte izquierda y parte derecha, es frecuente que estos sonidos estén de alguna forma divididos. Es normal escuchar un sonido partido de S2 causado por la contracción del ventrículo izquierdo y derecho en instantes de tiempo ligeramente diferentes. Vemos ahora un ejemplo en el que se aprecia una división importante en el sonido de S2 que produce una nueva componente S3. Así mismo una división de S1 produciría la componente S4.

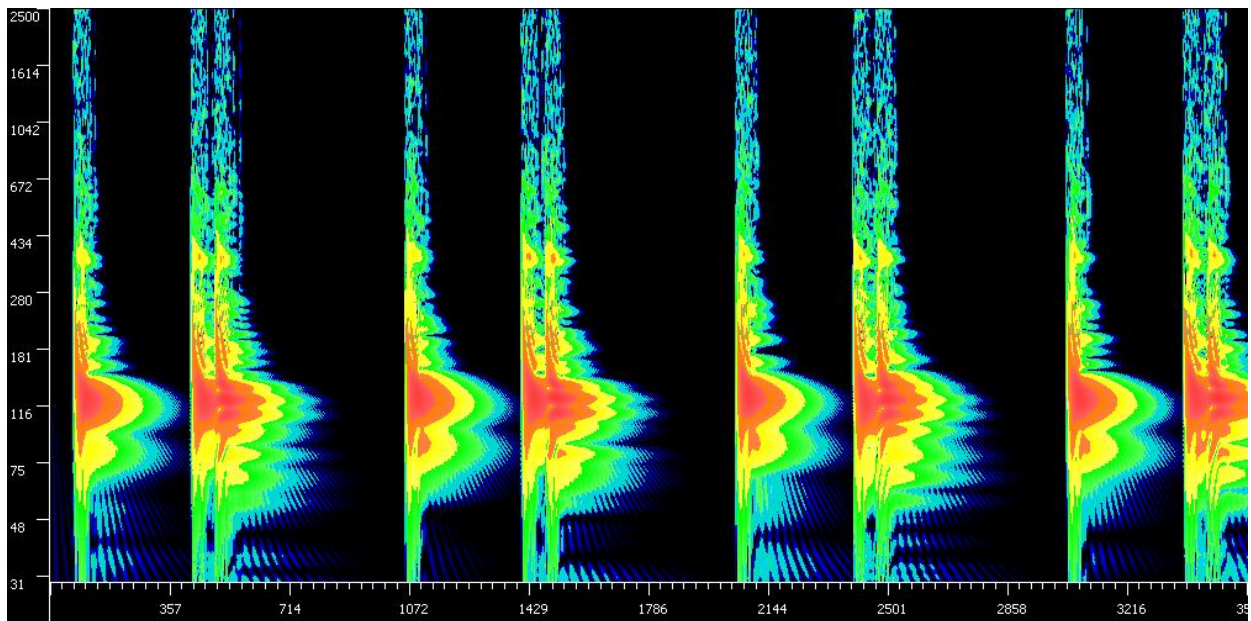


Figura 21: Análisis de un corazón con el sonido partido

Cuando una válvula está dañada el flujo de sangre produce un murmullo que se puede escuchar durante las normalmente fases silenciosas de la sístole o diástole. La estenosis aórtica es el estrechamiento u obstrucción de la válvula aórtica del corazón que no permite que ésta se abra adecuadamente, obstruyendo el flujo sanguíneo desde el ventrículo izquierdo a la aorta. En la siguiente figura vemos un ejemplo de esta anomalía.

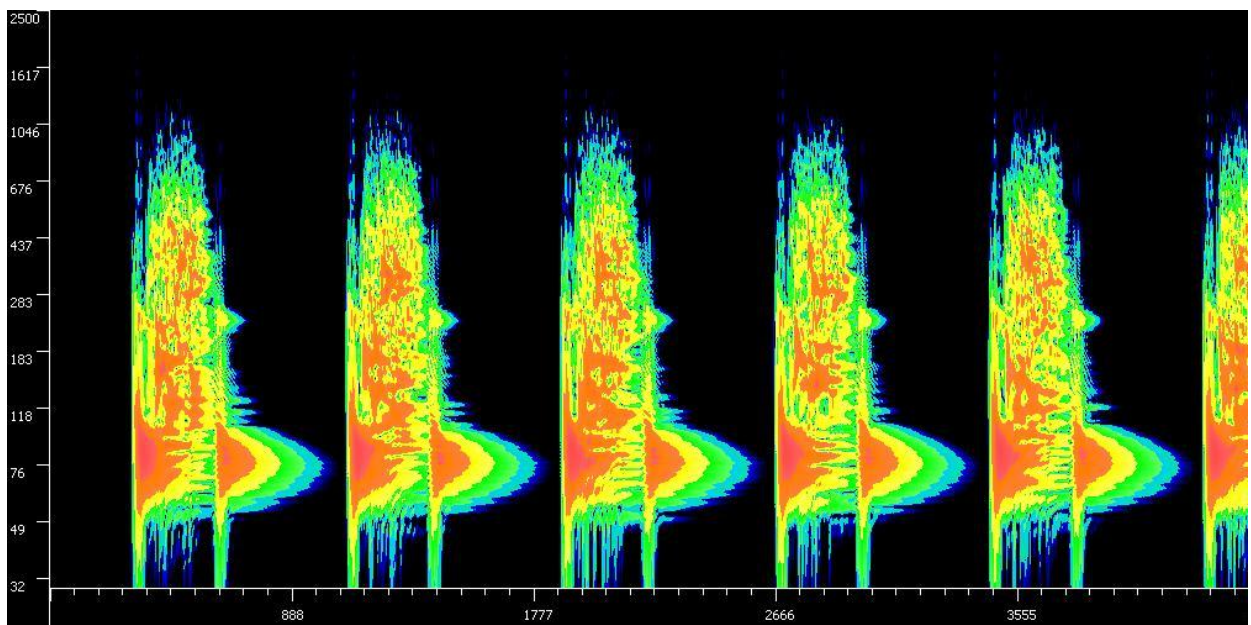


Figura 22: Análisis de sonido de un corazón con estenosis aórtica

Por último vamos a ver un ejemplo de regurgitación aórtica. La insuficiencia aórtica es una enfermedad de la válvula cardíaca en la cual la válvula aórtica se debilita o se abomba, impidiendo que dicha válvula cierre bien, lo cual produce un

subsecuente reflujo de sangre desde la aorta (el vaso sanguíneo más grande) hasta el ventrículo izquierdo (la cámara inferior izquierda del corazón). Este reflujo de sangre hacia el ventrículo izquierdo produce un murmullo característico cuyo análisis vemos a continuación.

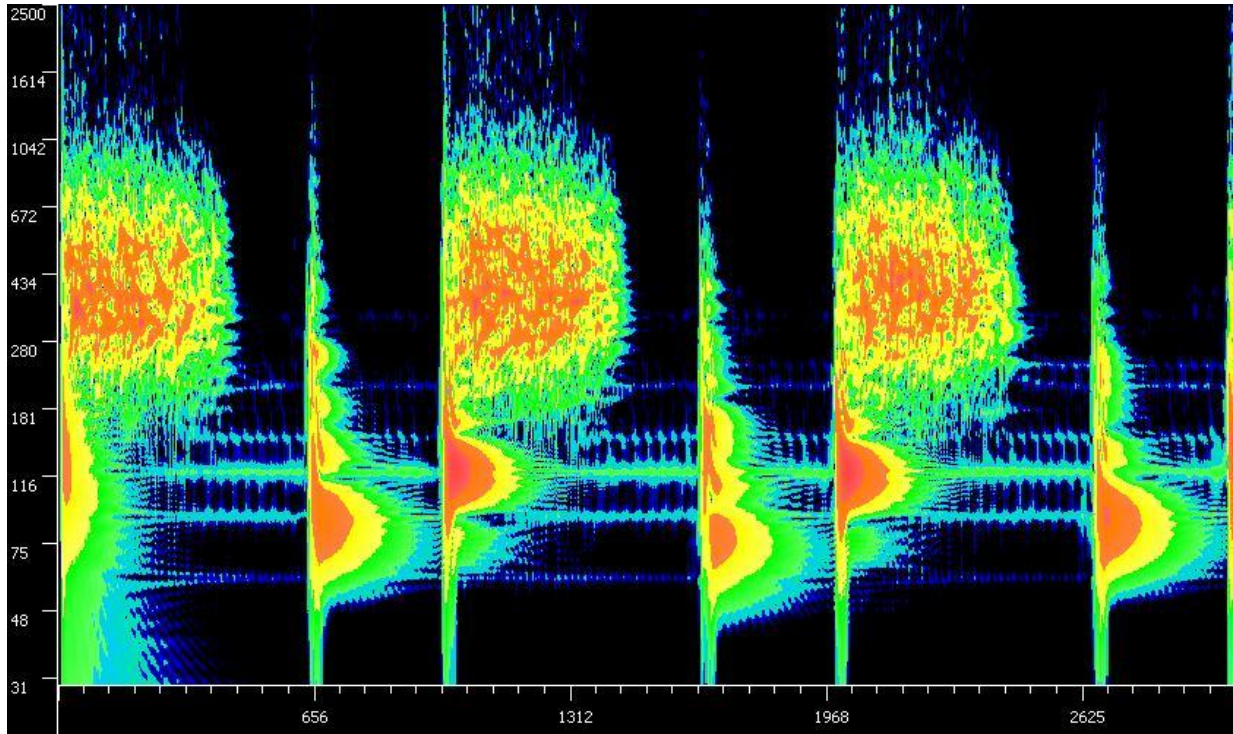


Figura 23: Análisis de sonido de un corazón con regurgitación aórtica

Podemos apreciar 2 sonidos claramente diferenciados en la diástole. En primer lugar el latido típico del corazón producido al cerrarse la válvula y por otro lado el murmullo que se produce por el reflujo de sangre a la ventrícula.

De los ejemplos anteriores podemos sacar las siguientes conclusiones:

- Cada uno de los componentes de sonido del corazón (S1-S4) puede ser modelado en detalle para su uso en posteriores comparaciones.
- Se puede usar el cocleograma para la comparación de latidos de distintas personas y poder buscar patrones comunes o incluso encontrar un latido medio.
- El análisis puede usarse como complemento al diagnóstico realizado a través del estetoscopio.

4.3.- ANÁLISIS DE SONIDOS PRODUCIDOS POR RUEDAS DENTADAS

CocleoMatic también puede utilizarse para el diagnóstico de posibles averías en maquinaria industrial por medio del análisis de los sonidos que producen. Vamos a ver un ejemplo sencillo en el que analizamos el sonido producido por una rueda dentada.

Se supone que las ruedas utilizadas en las grabaciones giran con una velocidad de 60 revoluciones por minuto (con lo cual en un segundo, da una vuelta completa).

Primero consideremos una rueda sin fallo:

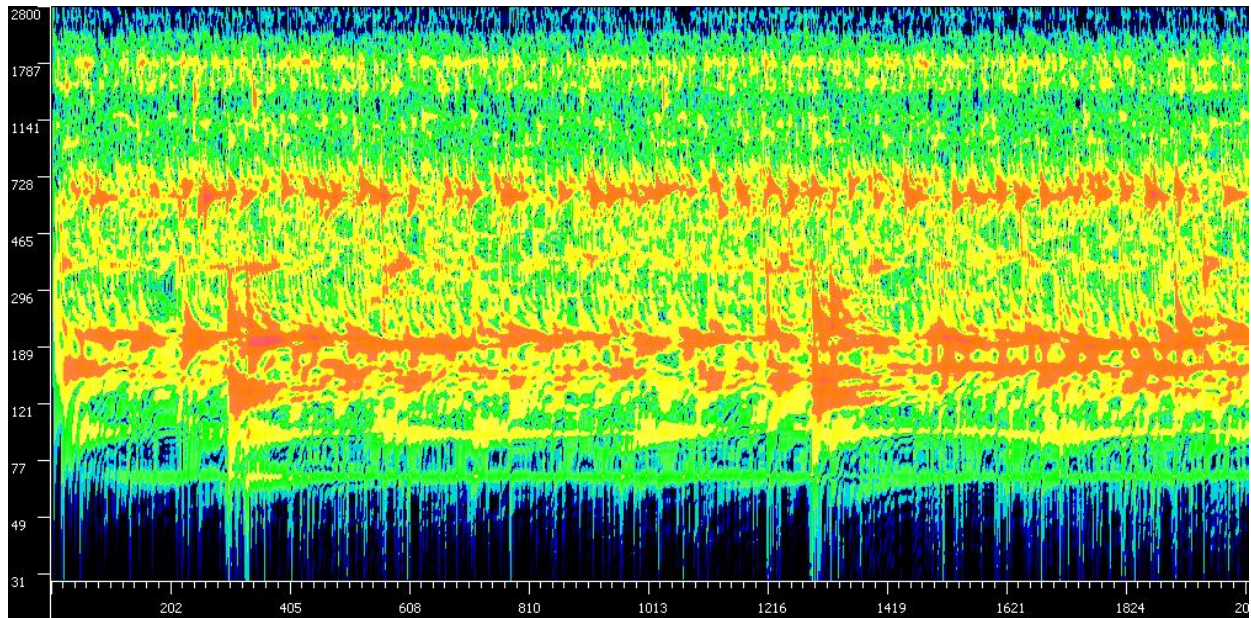


Figura 24: Análisis de sonido de una rueda dentada sin fallo

Se puede ver que el análisis contiene principalmente ruido, aunque sigue un patrón: Cada segundo se repite el análisis, es decir, es idéntico al análisis del segundo anterior (alguna pequeña diferencia puede ser debida a casi imperceptibles ruidos de fondo).

Veamos las diferencias con un análisis de un sonido de rueda dentada en el que varios dientes chocan con algo:

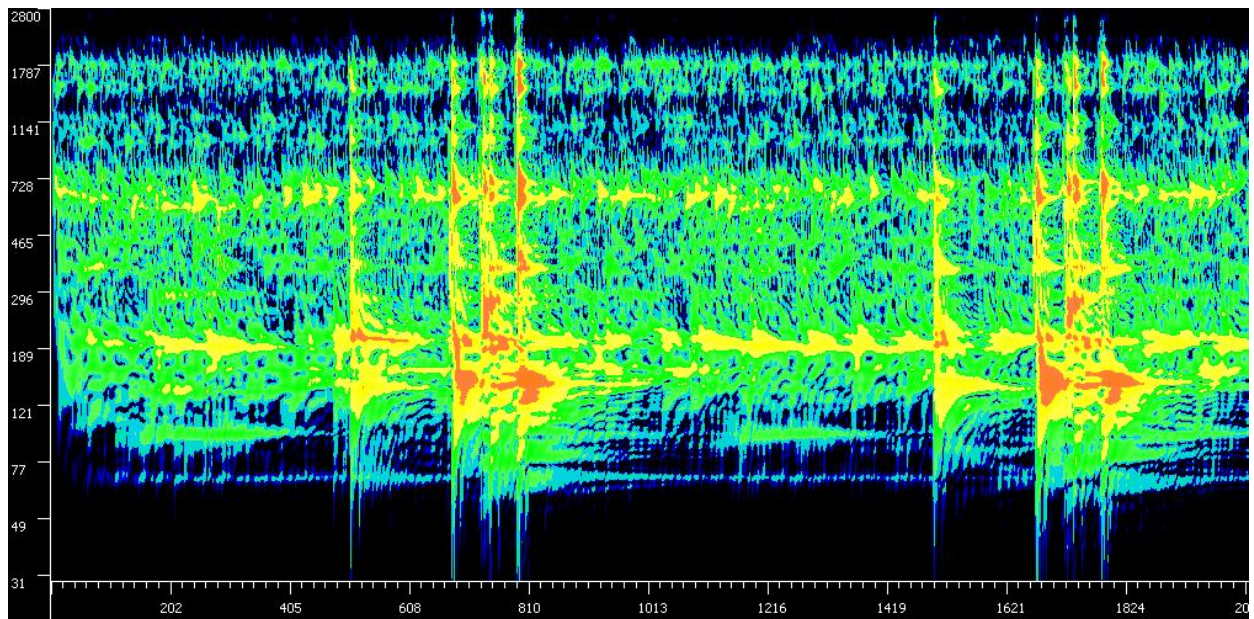


Figura 25: Análisis de sonido de una rueda dentada con choques

Los pulsos (líneas verticales) que se observan son causados por el choque de los dientes. Otra vez se observa que cada segundo el análisis se repite. El motivo por el que en este análisis sólo se aprecian magnitudes altas en los pulsos, es porque se amplificó la entrada de los dos análisis. Al ser la onda del primer análisis más "lineal" (no contiene pulsos apreciables), la amplificación resulta más evidente a lo largo de todo el sonido.

5.- ¿POR QUÉ COCLEOMATIC?

5.1.- ANÁLISIS DE FRECUENCIA VS. COCLEOGRAMA

El análisis de espectros o análisis de frecuencia se define como la transformación de una señal representada en el dominio del tiempo hacia la representación en el dominio de la frecuencia. Un análisis de espectro facilita por ejemplo la identificación de las fuentes de vibración no deseadas, o puede servir también para ecualizar las instalaciones de sonorización, compensando en tiempo real la respuesta en frecuencia de los equipos y altavoces según las características acústicas de un local, en las que influyen desde la absorción de los materiales hasta la ocupación de público.

Casi todas las técnicas existentes de análisis de espectro están basadas en la teoría de Fourier, tanto en su forma original como en posteriores modificaciones.

Jean Baptiste Fourier estaba trabajando para Napoleón durante la invasión de Egipto en un problema de sobrecalentamiento de cañones, cuando dedujo su serie, la mítica Serie de Fourier, aunque en un principio con el propósito de aportar una solución a la conducción de calor. A primera vista, podría parecer que hay una gran distancia entre cañones sobrecalentados y análisis de frecuencia, pero la realidad es que las mismas ecuaciones son aplicables en los dos casos. Fourier más tarde generalizó la Serie en la Transformada Integral de Fourier. La llegada del análisis de las señales digitales naturalmente llevó a la llamada Transformada Discrecional de Fourier y la Transformada Rápida de Fourier o FFT.

CocleoMatic tiene como principio el acercarse lo más posible a la manera en la que el oído humano entiende el sonido. No se trata por tanto de un análisis de frecuencia común, como el que pueden ofrecer la gran infinidad de analizadores de espectro disponibles, con enfoques completamente distintos.

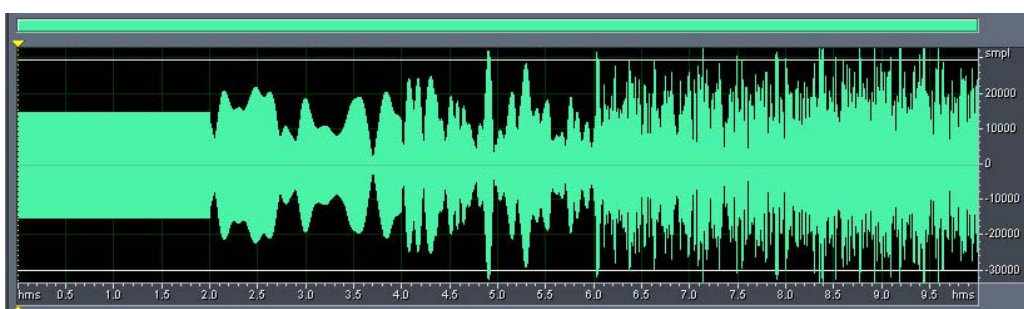


Figura 26: Forma de onda

La representación de una forma de onda, que podemos observar en la figura anterior, muestra la amplitud de la misma para cada instante de tiempo. Se trata de distintos tonos de dos segundos de duración, que se van sumando progresivamente, elevando su frecuencia.

El análisis de frecuencia de la forma de onda mostrada anteriormente presentaría el siguiente aspecto:

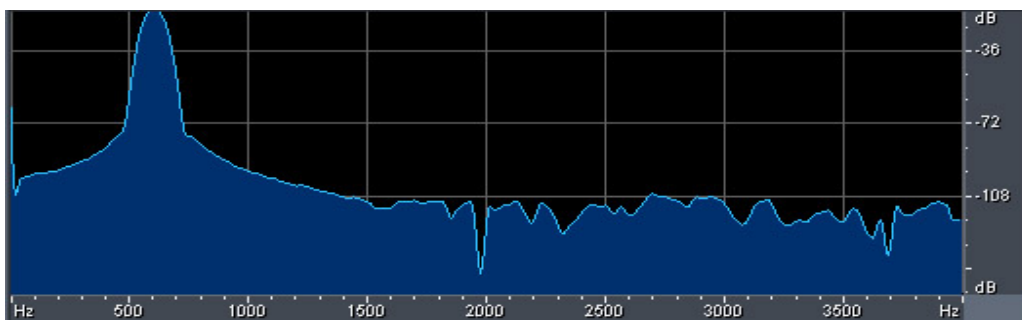


Figura 27: Espectro

El eje horizontal muestra las frecuencias presentes en el sonido, y el eje vertical la intensidad máxima con la que se manifiestan a lo largo del mismo.

El análisis que CocleoMatic efectúa no es estrictamente un análisis de frecuencia, sino que podría interpretarse como una representación en 3 dimensiones del mismo, en el que la intensidad de cada frecuencia se mide mediante una escala de colores.

El aspecto del cocleograma para el archivo estudiado sería el siguiente:

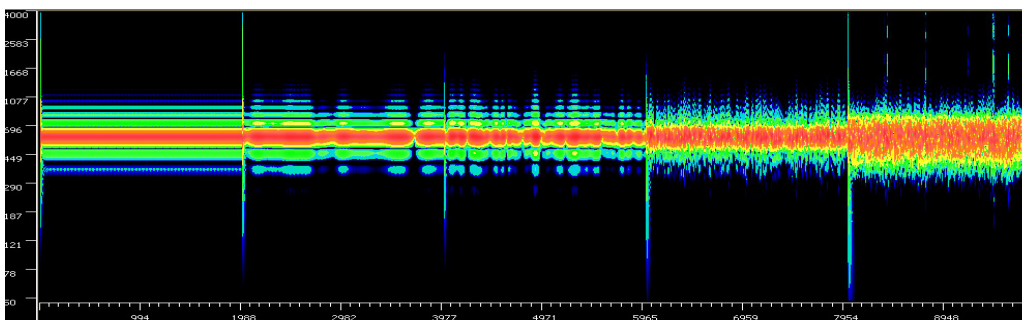


Figura 28: Cocleograma

El cocleograma se mantiene en el dominio del tiempo, representándose en el eje Y las distintas frecuencias audibles por la cóclea para cada instante mediante un color que indica el nivel de energía que alcanza su resonador correspondiente. Como ya se ha insistido anteriormente, lo que se muestra es lo que percibe la cóclea, y no un análisis purista del sonido real.

5.2.- VENTAJAS

CocleoMatic presenta claras ventajas frente a otros programas de uso similar, ya no sólo por su enfoque como simulador del oído humano, sino porque como analizador de frecuencia ofrece una precisión y posibilidades muy superiores a otros que gozan de reconocido prestigio.

5.2.1.- PRECISIÓN

Casi la totalidad de analizadores de espectro basan su análisis en el algoritmo de la FFT. La Transformada rápida de Fourier o FFT es un algoritmo para calcular la DFT de manera rápida y eficaz, y que fue desarrollado para adaptar la transformada discreta para su uso con computadoras digitales. El algoritmo posibilita un análisis rápido, pero limitado.

James W. Cooley y John W. Tukey fueron acreditados con el descubrimiento de la FFT en 1967, aunque ya existía anteriormente. El algoritmo pone algunas limitaciones en la señal y en el espectro resultante. Por ejemplo: la señal de la que se tomaron muestras y que se va a transformar debe consistir de un número de muestras que ha de ser potencia de dos. La mayoría de los analizadores FFT permiten la transformación de 512, 1024, 2048 o 4096 muestras. El rango de frecuencias cubierto por el análisis FFT depende de la cantidad de muestras recogidas y de la proporción de muestreo, e introduce discontinuidad en la señal.

Es importante mencionar que por la morfología del algoritmo utilizado, el tiempo de ejecución va a ser muy alto si lo comparamos con otros analizadores de espectro. El algoritmo de análisis de CocleoMatic no introduce ninguna limitación en la señal a costa de un incremento en el tiempo, pero dado que el enfoque de la aplicación es el análisis buscando altos niveles de precisión, no tiene sentido el estudio de archivos de sonido con longitud elevada, acción que por otra parte puede llevarse a cabo sin ningún problema excepto un aumento en el tiempo de espera.

Para ofrecer una aproximación del rendimiento, conste que CocleoMatic invierte en analizar un archivo de 10 segundos con una frecuencia de muestreo de 8000Hz 9.2seg (medida realizada en una maquina con procesador Intel Celeron a 1.6Mhz).

5.2.2.- PARAMETRIZACIÓN

Como ya se ha explicado con anterioridad, muchos de los fenómenos que intervienen en la interpretación del sonido por parte de la cóclea pueden variar, de manera que pueden ser modificados a la hora de realizar un análisis.

La dureza representa en el algoritmo de CocleoMatic el número de ciclos que necesita un resonador para alcanzar su máxima excitación, el número de resonadores limita el nivel de detalle y número de frecuencias que se excitan directamente, que siendo en el cuerpo humano aproximadamente de 15.000, ha quedado establecido en la aplicación con un rango entre 1 y 1000, posibilitando así una proporción suficiente para obtener resultados fieles a la realidad.

Además de los mencionados, existen otras posibilidades de personificación del análisis que amplían aún más su grado de precisión.

5.2.3.- ANÁLISIS PARCIAL VS. ZOOM

Los analizadores de espectro generalmente ofrecen la posibilidad de observar con detalle una parte del análisis completo. Es una característica básica y necesaria, pero que sin embargo resulta pobre en muchos de los programas que hemos contrastado.

CocleoMatic ofrece esta posibilidad, con la ventaja de que no se trata de una "ampliación" del análisis original, sino de un nuevo análisis restringiendo como límites del mismo, el origen y el final de la zona deseada. Es decir, no se trata de un Zoom al uso como el que encontramos en la mayoría de analizadores, sino de un recálculo para una porción concreta del sonido de entrada con las mismas características y precisión que si analizásemos el sonido completo.

5.3.- COMPARANDO COCLEOMATIC

Vamos a comparar diversos análisis de espectro, para intentar entender las diferencias entre unos y otros, y analizar las ventajas e inconvenientes que presenta CocleoMatic frente a software significativo dentro del campo que estamos estudiando.

Para ello, vamos a elegir dos programas lo suficientemente representativos, y que forman parte de dos vertientes claras del DSP hoy día: Cool Edit Pro y Baudline. El primero de ellos proviene de la Gran Industria, por y para un público amplio, tanto profesional como aficionado, y ha de mantener una buena imagen depurada durante años cumpliendo con la obligación de mejorar e innovar el sector. El segundo es un programa hecho por y para amantes del DSP, exclusivamente en LINUX, y sin más pretensión que mostrar un gran trabajo y ofrecer la posibilidad a cualquier usuario de beneficiarse de sus buenos resultados.

Cool Edit Pro 2.0 es uno de los editores de audio más utilizados en el mundo. Durante muchos años ha sido un referente para la edición y masterización de audio a nivel profesional y es uno de los programas más extendidos tanto en aplicaciones musicales como acústicas.

Desarrollado por Sintrillium Software, ha sido recientemente adquirido por Adobe para convertirse en el editor de audio oficial de la compañía. (<http://www.adobe.com/special/products/audition/syntrillium.html>)

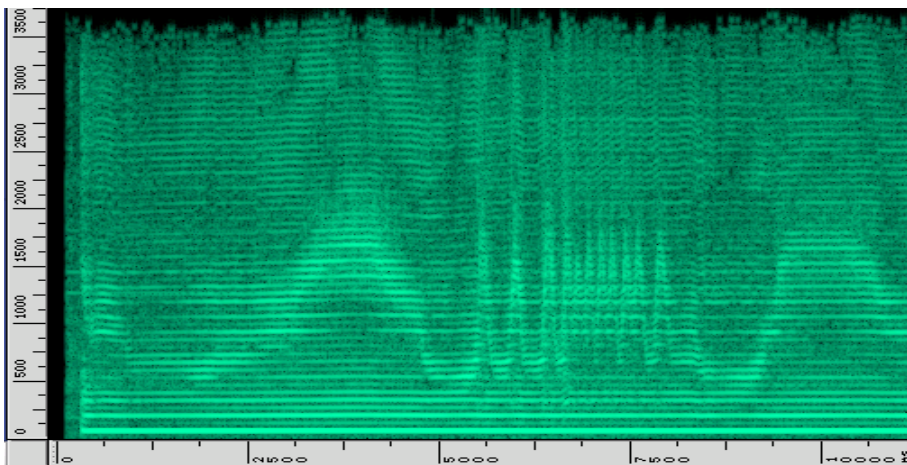
Ofrece un sinfín de posibilidades de edición y grabación multipista, así como de procesamiento digital de sonido. Entre sus prestaciones oficiales, presume de contar con un analizador de espectro de altas prestaciones.

Baudline es una herramienta de análisis de señal en tiempo real con un analizador de espectro muy similar en aspecto, que no en funcionamiento, a CocleoMatic.

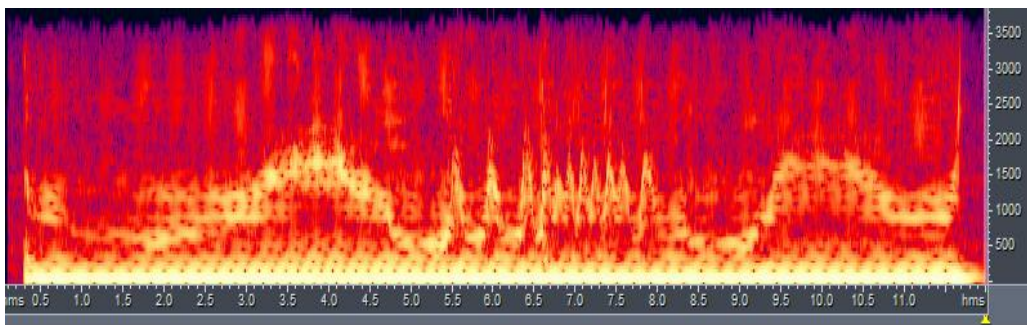
Está desarrollado por SigBlips, una empresa que trabaja, según sus palabras, "for the love of the signal". Ofrece resultados muy conseguidos, y está basado en la FFT. (<http://sigblips.com>, <http://www.baudline.com>)

- Analizando el mismo sonido de Didgeridoo con las tres aplicaciones:

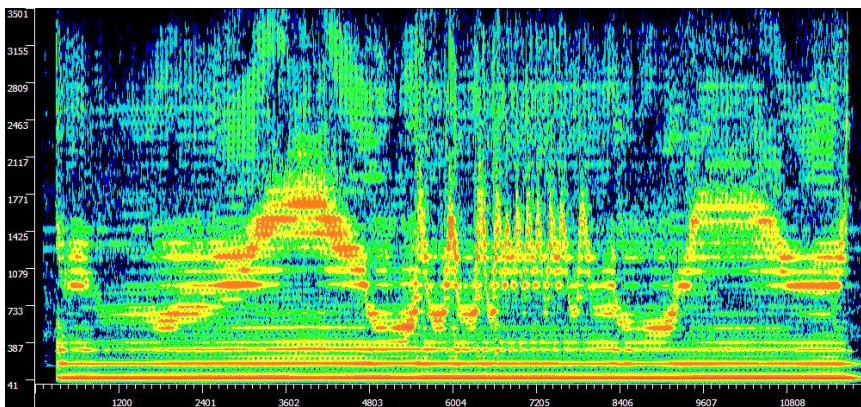
Baudline:



Cool Edit Pro 2.0:



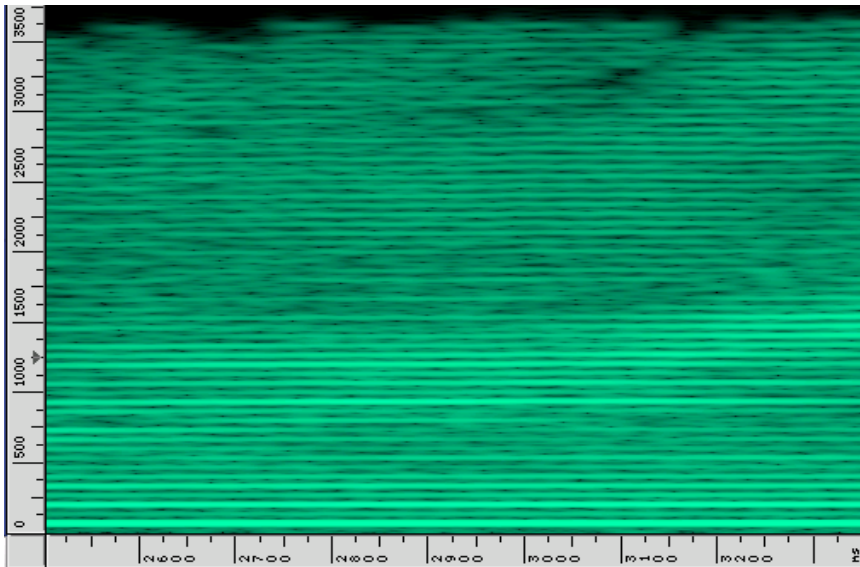
CocleoMatic:



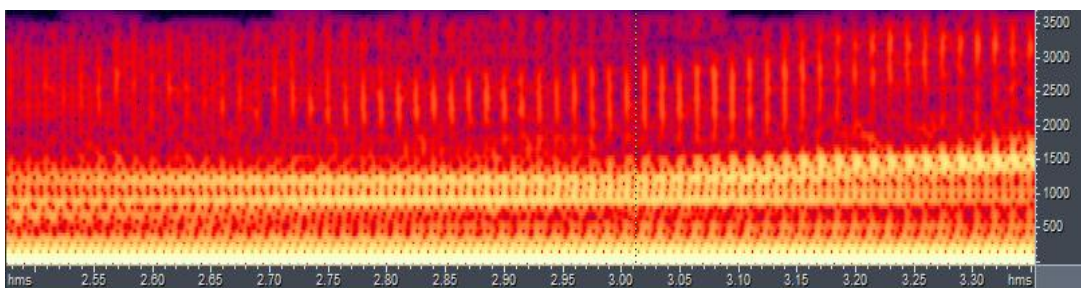
Ya a este nivel, se observan diferencias notorias en las líneas de armónicos más graves, quedando Cool Edit peor parado que los otros dos programas.

- Ampliando la misma sección del Didgeridoo anterior con los 3 programas:

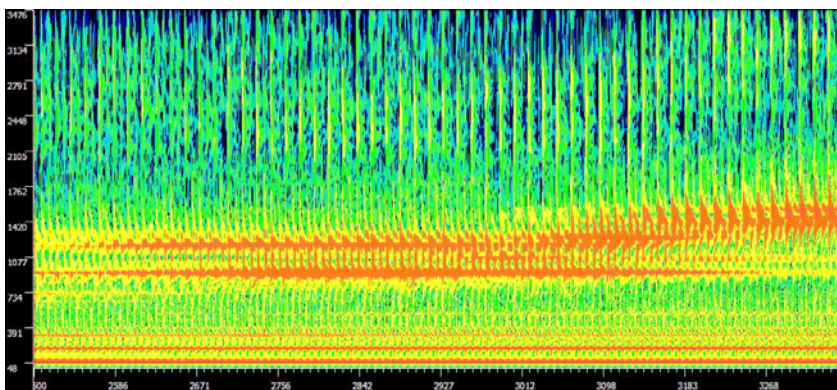
Baudline:



Cool Edit:



CocleoMatic:

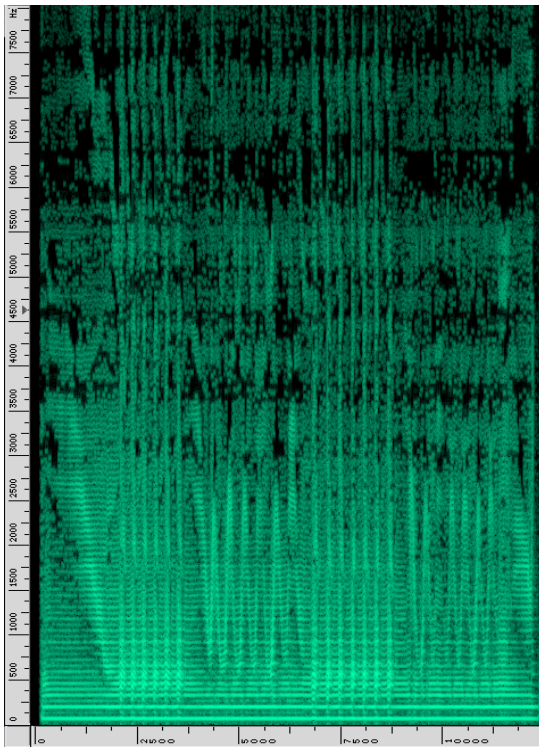


Se puede observar claramente como tanto en el caso de Baudline como en el de Cool Edit se trata de una ampliación a nivel gráfico y no analítico, perdiéndose precisión y haciendo difícil la distinción, por ejemplo, de los armónicos más bajos, en especial en Cool Edit, prácticamente imposible.

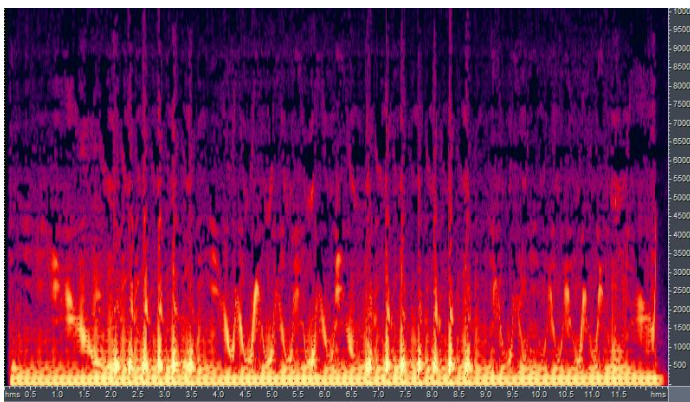
Las únicas posibilidades de personalización que aportan Baudline y Cool Edit están limitadas por la propia naturaleza de su motor de análisis, la FFT, pudiendo variar el tamaño de su ventana.

Observamos un caso más de análisis de DidgeriDoo, con similares conclusiones:

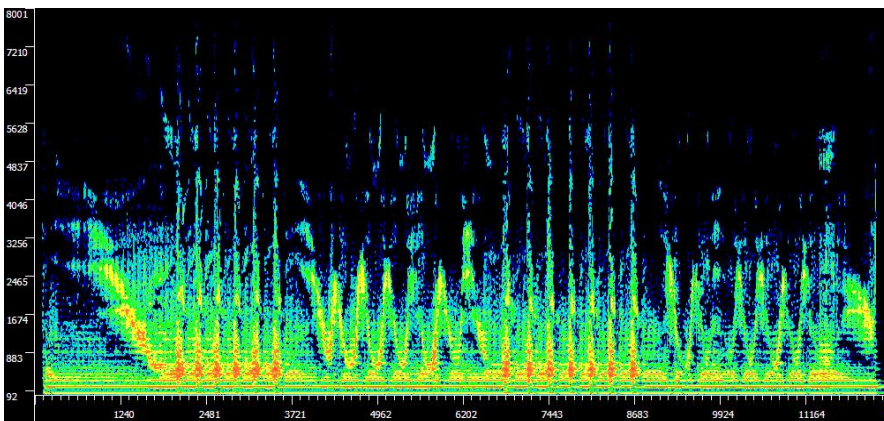
Baudline:



Cool Edit:

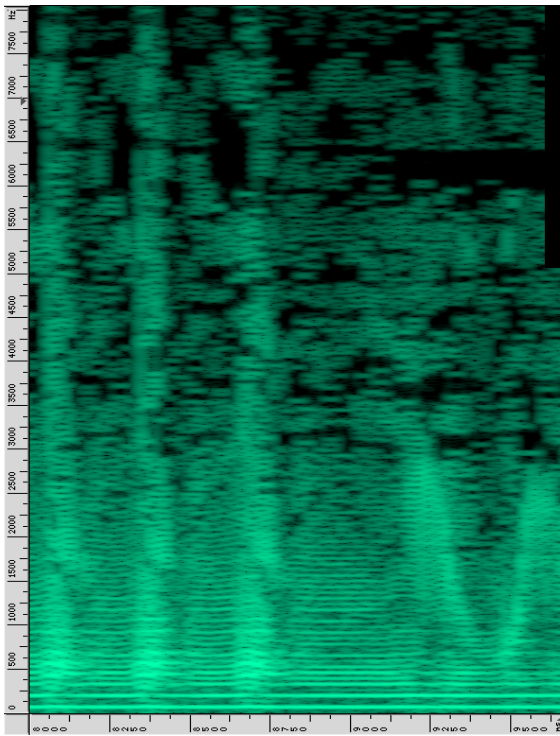


CocleoMatic:

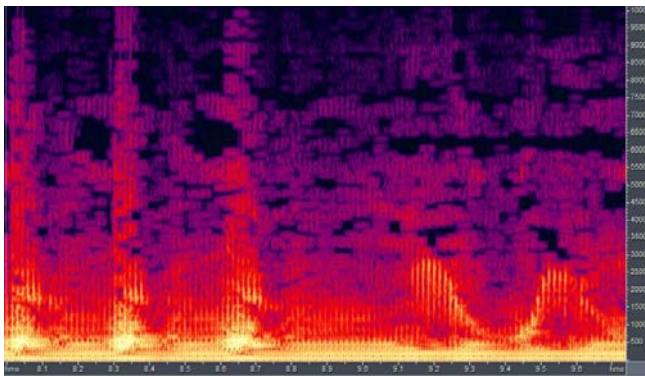


- Ampliando:

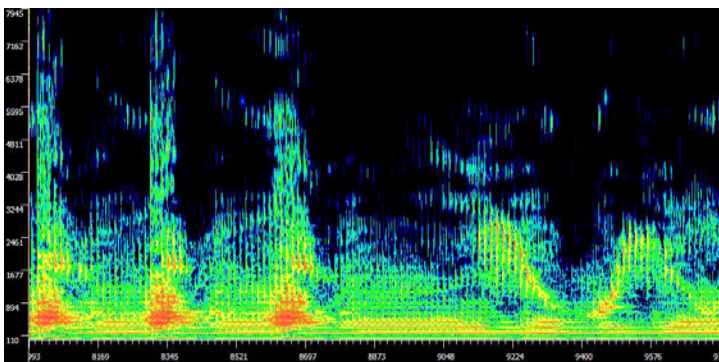
Baudline:



Cool Edit:



CocleoMatic:



6.- CONCLUSIONES

La experiencia de investigar con cualquier tipo de software enfocado a la informática musical nos pareció atractiva desde el principio, ya que es un área de difícil (por no decir imposible) estudio a lo largo de la carrera. Esto, sumado al hecho de que la idea original propuesta por nuestro director de proyecto, Jaime Sánchez, nos resultó muy atractiva y original hizo que nos decantáramos por ello con muy pocas dudas.

Para el desarrollo de la aplicación era necesario familiarizarse con nuevos conocimientos tanto a nivel técnico como teórico, en el sentido de que al margen de aprender a utilizar las tecnologías elegidas para la creación e implementación de la aplicación, ha sido necesario adquirir una base teórica del funcionamiento del oído humano, además de algunas nociones fundamentales de acústica y, algo más en profundidad, de análisis de Fourier. El hecho de que el ámbito de estudio no fuera exclusivamente la informática y que permitiese adentrarse también en otras áreas que nada tienen que ver con ella, ha sido otra de las oportunidades que nos ha brindado CocleoMatic, y de la que nos sentimos también muy orgullosos y agradecidos.

A nivel de objetivos, hemos cumplido tan sólo una parte de los pretendidos, en el sentido de que el algoritmo en el que se basa el análisis ofrece muchas posibilidades, y quizás porque forma parte de ese ámbito que para nosotros era hasta ahora totalmente desconocido y que se encuentra fuera del alcance de la titulación que cursamos, quizás porque nuestra organización no ha sido todo lo eficiente que debería, o simplemente porque quizás no hemos sido capaces de sacarle todo el partido que nos hubiera gustado, han quedado fuera de la versión final algunas de las mejoras y experimentos que queríamos incluir y que hemos considerado no se encontraban a estas alturas lo suficientemente desarrollados como para formar parte de una aplicación que pretende ofrecer posibilidades robustas y contrastadas.

En cualquier caso, desde el comienzo teníamos claro que fuera cual fuera el punto hasta el que consiguiésemos llegar, se intentaría en la medida de lo posible que el aspecto de la aplicación fuera lo más atractivo y profesional posible, que las prestaciones ofrecidas no tuviesen puntos débiles, y que nuestro software se diferenciase del resto de programas similares por aportar una visión nueva y diferente de las que ya existen. Y creemos firmemente que es algo que hemos conseguido. Estamos muy satisfechos con el resultado estético de CocleoMatic, a pesar de la cantidad de dificultades que hemos encontrado en la utilización de las librerías gráficas elegidas, y estamos muy satisfechos de todas y cada una de las funciones que ofrece.

Hemos contrastado CocleoMatic con prestigiosos programas que realizan acciones similares, y creemos que estamos en condiciones de afirmar que no tiene nada que envidiar a gigantes del software musical, si bien incluso supera a muchos de ellos en precisión y posibilidades.

Existen muchísimas ampliaciones y mejoras posibles, algunas de ellas en desarrollo y que como ya hemos mencionado, no creíamos lo suficientemente avanzadas como para formar parte de la versión final descrita en esta memoria. Y si

bien tenemos la intención e ilusión de intentar incluirlas en un futuro inmediato en la versión actual, sabemos que si no somos nosotros, la cuarta persona detrás de este proyecto lo intentará (y no hablamos de Max Unger).

Como apunte final, queremos agradecer a Jaime Sánchez el habernos confiado su idea, y el habernos servido de guía en este intento de darle forma.

BIBLIOGRAFÍA

- “Mathematics and music”, Dave Benson, Ed. 2004
- <http://www.geo.cornell.edu/geology/classes/brown/eas434/Notes/Fourier%20family.doc>
- <http://www.earlevel.com/Digital%20Audio/FFT.html>
- http://www-gpi.physik.uni-karlsruhe.de/pub/jmann/Skripte/MGdG_Skript.pdf
- <http://glade.gnome.org/>
- http://valbuena.fis.ucm.es/~adame/programacion/Glade/manual/Manual_de_glade.pdf
- <http://www.gtk.org/>
- <http://egeneralmedical.com/egeneralmedical/listohearmur.html>
- <http://www.wilkes.med.ucla.edu/inex.htm>
- <http://www.nlm.nih.gov/medlineplus/ency/article/003266.htm>
- <http://www.ai.rug.nl/~tjeerd/CPSP/>
- http://www.anarkasis.com/pitagoras/770_contexto_disonancia/index.htm
- <http://www.fisio.buap.mx/online/-COCLEA%202003%20Formateado%20b.htm>
- <http://www.people.cs.uchicago.edu/~odonnell/OData/Courses/CS295/perception.html>
- <http://www.dliengineering.com/vibman-spanish/glennspanol-toc.htm>
- <http://www.bores.com>
- <http://www.dspdimension.com>

APÉNDICE 1: TABLA DE FIGURAS

Figura 1: Gráfica de Fletcher y Munson	- 8 -
Figura 2: El oído humano.....	- 9 -
Figura 3: Esquema del oído medio e interno	- 10 -
Figura 4: Cóclea desenrollada.....	- 10 -
Figura 5: Fenómeno del enmascaramiento	- 12 -
Figura 6: Aspecto de la ventana principal de un proyecto Glade	- 19 -
Figura 7: Aspecto de la paleta de widgets.....	- 20 -
Figura 8: Aspecto de la ventana del árbol de widget	- 21 -
Figura 9: Aspecto de la ventana de propiedades.....	- 21 -
Figura 10: Aspecto de la interfaz de la aplicación	- 28 -
Figura 11: Diálogo Cambiar Parámetros	- 30 -
Figura 12: Ventana del CocleoGrabador	- 31 -
Figura 13: Ventana Configurar reproductor.....	- 31 -
Figura 14: Ventana Propiedades de archivo	- 32 -
Figura 15: Análisis de sonido de guitarra 1	- 34 -
Figura 16: Análisis de sonido de guitarra 2	- 35 -
Figura 17: Análisis de sonido de didgeridoo 1	- 36 -
Figura 18: Análisis de sonido de didgeridoo 2	- 36 -
Figura 19: Las partes del corazón	- 37 -
Figura 20: Análisis de sonido de un corazón sano.....	- 38 -
Figura 21: Análisis de un corazón con el sonido partido.....	- 39 -
Figura 22: Análisis de sonido de un corazón con estenosis aórtica.....	- 39 -
Figura 23: Análisis de sonido de un corazón con regurgitación aórtica.....	- 40 -
Figura 24: Análisis de sonido de una rueda dentada sin fallo.....	- 41 -
Figura 25: Análisis de sonido de una rueda dentada con choques.....	- 42 -
Figura 26: Forma de onda	- 43 -
Figura 27: Espectro	- 44 -
Figura 28: Cocleograma.....	- 44 -

APÉNDICE 2: AUTORIZACIÓN

El presente documento sirve para mostrar la total conformidad de los tres componentes del grupo de Sistemas Informáticos del curso 2004/2005 que ha desarrollado el proyecto de fin de carrera denominado "Analizador de un espectro musical" en **autorizar a la Universidad Complutense para difundir y utilizar con fines académicos, no comerciales, y mencionando expresamente a sus autores, tanto esta memoria como el código, la documentación y/o el prototipo desarrollado.**

Para garantizar dicha autorización, a continuación sirva la identificación y firma de cada uno de los componentes de este grupo de trabajo:

Andrés Jiménez Sánchez
DNI N°: 11846198 W
Firma:

Amaia de Miguel Morate
DNI N°: 70057347 Z
Firma:

Javier Villellas Bernal
DNI N°: 51077049 Y
Firma: