Detector de notas musicales y permutador de instrumentos.

Estanislao Claucich, Axel Alberto Gorostidi y Juan Cruz Leiva
Universidad Nacional del Litoral. eclaucich@hotmail.com, gorounl@gmail.com, juancojcl@hotmail.com

***Resumen* — Se busca identificar notas musicales en el tiempo, provenientes de una entrada de audio generada por un solo instrumento particular y sintetizar el audio de salida reproducido con un instrumento diferente al de entrada. La principal motivación para realizarlo es la posibilidad de cambiar instrumentos de una canción de forma virtual, sin la necesidad de volver a producirla físicamente con otros instrumentos.**

Para lograr este propósito, se utiliza un banco de muestras de notas musicales, generadas con un sintetizador digital que refleja notas individuales de varios instrumentos. A partir de las notas que sean detectadas por el algoritmo, se realiza una búsqueda en este banco utilizando las muestras del mismo para generar una nueva salida.

La propuesta consiste en un análisis por ventanas donde para cada una se obtiene su espectro en potencia. A partir de éste, se utiliza el algoritmo MHPS para identificar las notas existentes en la ventana, al cual se le realiza un postprocesamiento. Y, con la información obtenida de cada nota identificada, se genera la nueva salida.

***Palabras clave* — intercambio instrumento, detección de notas, MHPS.**

I. Introducción

El reconocimiento de notas musicales provenientes de un audio consiste en identificar las frecuencias que caracterizan a éste a través del tiempo.

Las frecuencias en las que trabajan los instrumentos es muy amplia, y cada uno generará un pitch distinto, es decir que el humano tendrá una percepción distinta de estas frecuencias según el instrumento.

Los instrumentos, al sonar una nota, generarán a su vez una serie de armónicos correspondientes a la frecuencia de esta nota. La identificación de estos armónicos será de vital importancia para la futura detección de las mismas.

Se propone un algoritmo que mejora los resultados obtenidos por el algoritmo de detección de notas MHPS, a través de un post-procesamiento de su salida.

El algoritmo consta de una serie de parámetros, donde sus valores afectarán en gran medida a los resultados obtenidos. Es por esto, que se realiza una identificación de los mejores parámetros para el mismo, mediante prueba y error. Los resultados obtenidos con la variación de estos parámetros se pueden observar en la sección *Resultados* de este documento.

A su vez, para la síntesis de audio, se proporciona un banco de muestras de distintas notas generadas por distintos instrumentos. Mediante las notas identificadas por el algoritmo propuesto, se realiza una búsqueda en este banco, para reproducir la nota correspondiente.

Hay muchos factores que pueden afectar al resultado final sintetizado, entre estos se encuentran las diferencias que existen en la generación de sonidos por los distintos instrumentos, junto con la cantidad de armónicos que puede generar cada uno, así como la potencia del instrumento en cada una de las notas.

Estos problemas son controlados en cierta medida por la propuesta realizada. Sin embargo, se mencionarán los problemas existentes en dicha propuesta, y lo necesario para seguir trabajando en la misma.

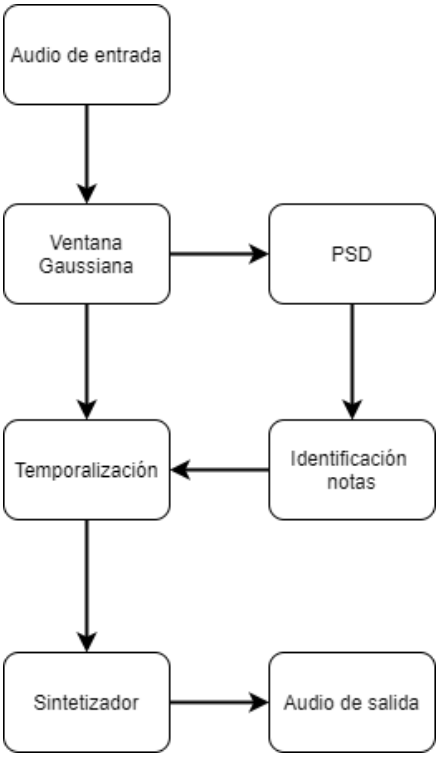

Fig. 1. Diagrama de bloques de sistema propuesto.

II. Materiales

Se generaron una serie de audios de pruebas con diferentes tipos de piano y guitarras. Creando una pequeña base de datos para realizar de una manera eficaz la comprobación y realización del algoritmo.

Para la síntesis de audio, se utilizaron muestras de audio de un segundo de duración con una amplia selección de notas para diferentes instrumentos. De esta manera, el algoritmo genera a partir de las mismas un banco de instrumento donde luego se podrán obtener muestras de las notas de los mismos para poder ser sintetizadas en el audio de salida. Esto permite que el audio generado, sea reproducido por uno o varios instrumentos diferentes al original.

III. MÉTODOS

A. Análisis por ventanas

La solución propuesta comienza realizando un análisis por ventanas a una señal de audio correspondiente con una melodía musical. La elección del ancho de estas ventanas será crítica para la eficiencia del algoritmo, ya que es necesario tener una buena resolución temporal para poder conocer con precisión en qué momento una nota comienza y termina de sonar, pero también, se necesita una buena resolución frecuencial para poder detectar las diferentes notas. Para esto se incluye el concepto de una frecuencia mínima donde, por debajo de ésta, el algoritmo no puede garantizar la correcta detección de las notas. De esta manera, se podrá ignorar la resolución frecuencial por debajo de esta frecuencia mínima, si es conocido previamente que la melodía en cuestión no trabaja en esta zona, permitiendo así que las ventanas sean de un ancho menor, mejorando la resolución temporal.

Otro concepto introducido es el de redundancia. El cual hace referencia al solapamiento existente entre las ventanas. Nuevamente, en la sección de *Resultados* se puede visualizar el impacto de este parámetro en el resultado del algoritmo.

El tipo de ventanas utilizadas son Gaussianas, ya que arrojan mejores y más fieles resultados, sin alterar demasiado el espectro en frecuencias de la señal correspondiente a la ventana. Esta elección, demanda más recursos que otro tipo de ventanas, pero son adecuadas para el trabajo ya que el análisis frecuencial es una parte indispensable del mismo.

B. Análisis espectral

Para cada ventana, se procede a calcular la Transformada de Fourier de la información contenida en las mismas.

Con el fin de mejorar la eficiencia del algoritmo, se aplica a cada una de estas transformadas una Ponderación A [1], la cual es una función muy utilizada en sonidos producidos por instrumentos, donde tiene en cuenta el volumen relativo percibido por el oído humano, ponderando las frecuencias en relación con esta percepción. A partir de inspecciones, se determinó que, utilizando ésta función, el posterior cálculo del espectro en frecuencias es más representativo, ya que las notas de mayor interés se verán más acentuadas en el mismo.

A partir de esta ponderación, se calcula la Densidad Espectral en Potencia (PSD – Power Spectrum Density) [3]. Este cálculo se realiza, en primera parte, para resumir la serie de Fourier (compleja) en valores reales; y en segunda parte, el PSD permitirá determinar la distribución de la varianza de la información en el tiempo, en el dominio frecuencial en componentes espectrales, en los cuales la señal podrá ser descompuesta. Como se verá más adelante, obtener una mejor representación de la distribución del espectro frecuencial es esencial para los cálculos futuros.

C. MHPS

Uno de los algoritmos más importantes utilizados en el trabajo corresponde al HPS (Harmonic Product Spectrum). La densidad espectral en potencia que se obtiene anteriormente, contiene información de la frecuencia fundamental de las notas y sus armónicos. El HPS tiene como objetivo, a partir de esta información, extraer la frecuencia fundamental de cada una, en base a un factor R el cual indica la cantidad de armónicos a considerar para cada nota y realizando submuestreos en la señal (Figura 2).

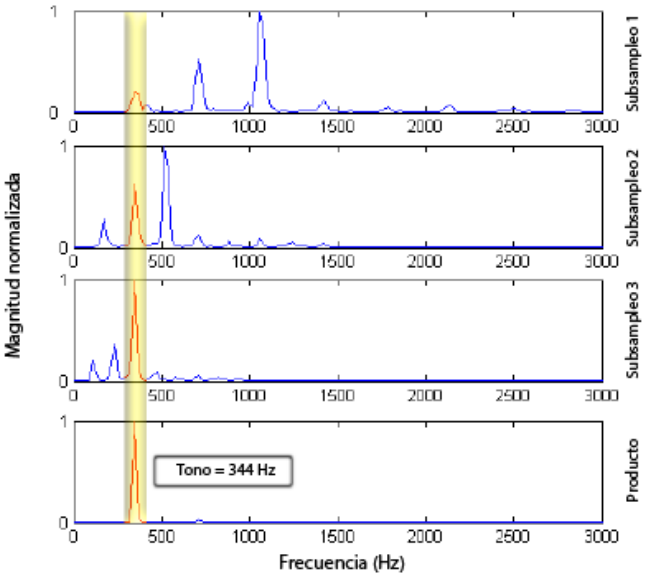

Fig. 2. Principio de funcionamiento de HPS.

Estando posicionado sobre un armónico de una nota musical (Figura 2 – Subsampleo 1), se puede conocer la frecuencia del próximo simplemente multiplicando la suya por dos. Por lo que, al comprimir el espectro varias veces, mediante el submuestreo, en cada iteración del mismo se puede comparar la señal resultante con la original y observar que los armónicos se alinean, es decir, el primer armónico de la señal original coincide en posicionamiento con el segundo de la submuestreada, el segundo con el tercero y así sucesivamente. Al seguir comprimiendo la señal, los armónicos de cada submuestreo coinciden con los del anterior. Al multiplicar los diversos espectros, se forma un pico muy claro en las frecuencias fundamentales (Figura 2 – Producto), permitiendo conocer su valor con cierta exactitud.

El factor "R" al que se hace referencia, indica la cantidad de armónicos necesarios que tiene que contener una nota en la señal, para poder ser considerada como tal. En el caso de la Figura 2, R=3. Esto es importante, ya que, si se elige un valor bajo, por ejemplo 2, se podrían estar detectando como notas, lo que en realidad son armónicos de alguna nota. Lo recomendable es un valor de 4 o 5, ya que suelen ser la cantidad de armónicos con energía relevante de una nota. En base a este valor, es la cantidad de submuestreos que se le realiza a la señal original, buscando en cada uno la alineación de un armónico con la frecuencia fundamental.

El algoritmo descripto anteriormente, presenta el error de detectar, generalmente, una nota una octava por encima de la real. Por esto, se utiliza un algoritmo adicional denominado MHPS (Modified Harmonic Product Spectrum) [2], que propone un mecanismo de competencia entre las notas detectadas anteriormente.

Para realizarlo, se parte del resultado del HPS, donde se analiza cada una de las notas y se las evalúa en base a un valor de tolerancia con respecto a su potencia local. Aquellas que superan esta condición, serán consideradas como notas. En caso contrario, se les da otra oportunidad en la cual se las vuelve a analizar frente a una nueva tolerancia. Este procedimiento se realiza tres veces, con distintas tolerancias cada vez menores.

Al finalizar el mecanismo de competencia, se obtienen las notas buscadas, con una mayor exactitud y con menos falsos positivos que el resultado original del HPS.

D. Post procesamiento MHPS

Sin embargo, este algoritmo, por más que soluciona los problemas entre octavas del HPS, no contempla otros falsos positivos de éste, que se deben a la identificación de dos o más frecuencias, de valores muy similares, como notas diferentes y a la introducción de frecuencias indeseadas en la fase de ventaneo.

Las notas musicales no se encuentran distribuidas *aleatoriamente* en el espectro de frecuencias, sino que siguen una regla, una función, que las posiciona en este espectro. La separación entre dos notas siempre estará dada por la relación $\sqrt[12]{2}$. Por lo tanto, si tomamos una frecuencia F_0 (la cual diremos que es una nota), sería incorrecto identificar frecuencias menores a $F_0 \cdot \frac{1}{2}(1 + \sqrt[12]{2})$ como notas distintas a F_0. Es decir, que F_0 estará trabajando en un intervalo dado por (1).

$$\left[F_0 \cdot \frac{1}{2}\left(1 + \frac{1}{\sqrt[12]{2}}\right), \quad F_0 \cdot \frac{1}{2}\left(1 + \sqrt[12]{2}\right)\right] \qquad (1)$$

Cada vez que se hable de energía o potencia de una nota, se habla de la energía o potencia concentrada en este intervalo.

Para resolver el problema de notas símiles, proponemos un algoritmo que, a partir de la salida del mismo, irá agrupando las notas *similares* en un mismo grupo. Dos notas serán similares, si una se encuentra dentro del intervalo de trabajo de la otra. De esta manera, se van creando grupos de notas a lo largo de todas las ventanas.

Ahora, para cada grupo formado, podrá existir (en general siempre sucede) más de una frecuencia contenida en el mismo. Entonces, se debe elegir aquella frecuencia que representa en mejor manera a todo el grupo. Para esto, se mide la potencia puntual de cada nota, y se determina que aquella que posee la mayor potencia puntual, será la representante de todo el grupo. Eliminando todas las demás frecuencias del mismo.

Gracias a este algoritmo, la cantidad de notas identificadas como falsos positivos se reduce considerablemente, como se detalla en la sección *Resultados*.

E. Temporalización

Hasta ahora solo se tenía en cuenta la información frecuencial de la señal, donde se obtuvieron las notas de la misma. A partir de éstas, se tiene como objetivo, añadir a cada una, su información temporal. Esta información constará del tiempo en el que la nota comenzó a sonar, y el tiempo donde ésta finalizó. Ambos medidos en relación a la señal completa del audio de entrada.

La propuesta realizada para esta sección, consta de una primera parte donde, a partir de conocer la última nota que se ha identificado en tiempo determinado, identificar si la siguiente nota corresponde a la misma o es una nueva. Básicamente, identificar si la nota sigue sonando o ya ha dejado de sonar. En caso de que se trate de una nueva nota, la nota anterior se marcará como “finalizada”, determinando su tiempo final, y a su vez determinado el tiempo inicial de esta nueva nota. En caso contrario, si la nota es la misma a la anterior, se marca a la misma como “sonando”. De esta manera, para cada nota identificada, se obtiene el tiempo donde comenzó y finalizó de sonar.

La segunda parte de esta propuesta, consiste en determinar cuándo una nota identificada como “sonando” pasa a identificarse como “finalizada”. Para esto, se propusieron dos tolerancias distintas. Una de estas se basa en la cantidad de ventanas contiguas en las que la nota está sonando. Es decir que, al haber identificado una nota, si durante X cantidad de ventanas, no se vuelve a encontrar, ésta será considerada como “finalizada”. La otra tolerancia, es utilizada para controlar la longitud temporal mínima de cada nota. Si una nota es considerada como “finalizada”, pero su duración es menor que esta tolerancia, será removida de esta categoría, y considerada como ruido.

Al finalizar el análisis de todas las ventanas necesarias para la señal completa de audio de entrada, se tendrá un conjunto de notas identificadas, donde para cada una se conocerá su frecuencia, su tiempo inicial y su tiempo final. Con esta información, ya es posible sintetizar el audio, y generar diferentes gráficas que permitirán visualizar el comportamiento del algoritmo propuesto.

IV. Resultados

Los resultados obtenidos a lo largo del algoritmo dependerán de ciertos parámetros mencionados anteriormente. En principal, se ve la diferencia producida al variar tanto la redundancia utilizada como la frecuencia mínima en cuestión.

En la Figura 3 se puede visualizar, para un conjunto de seis muestras de audio distintas, cómo varía la eficiencia del algoritmo al aumentar la redundancia del mismo. Recordar que, al aumentar la redundancia, aumenta la cantidad de ventanas totales que se utilizarán, aumentando la información de análisis, por lo que habrá más oportunidades de eliminar falsos positivos.

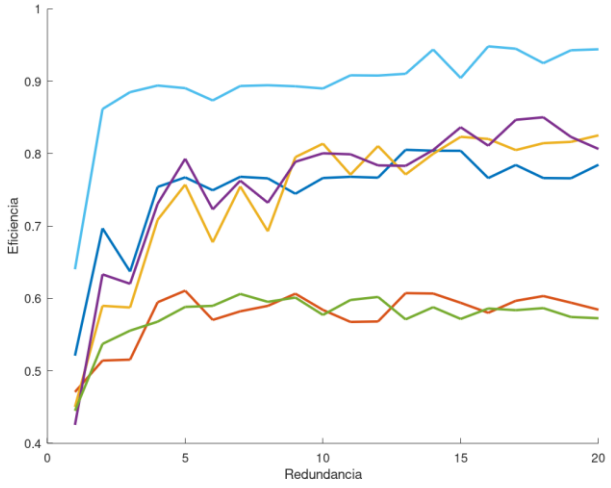

Fig. 3. Eficiencia en función de redundancia.

A su vez, en la Figura 4 se muestra, para las mismas seis muestras que en el caso anterior, cómo afecta la variación de la frecuencia mínima en la eficiencia del algoritmo. Al aumentar este parámetro, las ventanas serán cada vez de un ancho menor, perdiendo resolución frecuencial. Por lo tanto, dependiendo de la muestra utilizada, llegará un punto donde

esta resolución no permita identificar ninguna nota en dicha muestra.

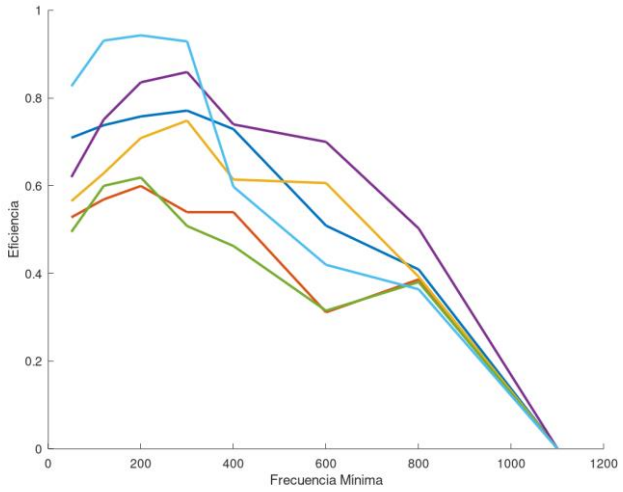

Fig. 4. Eficiencia en función de frecuencia mínima.

Para calcular la eficiencia del algoritmo, se tienen en cuenta tres tipos de errores, los cuales son calculados realizando una comparación directa entre el resultado del algoritmo propuesto y el MIDI de cada una de las muestras, el cual posee la información que se considerará como correcta.

El primero de estos errores (EFP – Error Falsos Positivos) equivale al porcentaje de falsos positivos sobre el total de notas en el MIDI.

Otro de los errores (ETFP – Error Temporal Falsos Positivos) se determina como la relación entre el tiempo total de los falsos positivos y el tiempo total del audio de muestra.

Y, por último, un error (ETNC – Error Temporal Notas Correctas) que se determinará como la mediana de los errores temporales de las notas acertadas correctamente.

Finalmente, el error total obtenido será como se muestra en (2). Donde α, β y μ, se corresponden con los pesos de cada uno de los errores. Según lo que uno exija del algoritmo estos pesos podrán variar. Es decir que, si es de mayor interés que las notas sean consideradas correctamente, sin importar sus errores temporales, α será más grande que β y μ.

$$Error = \frac{\alpha * EFP + \beta * ETFP + \mu * ETNC}{\alpha + \beta + \mu} \qquad (2)$$

Para los resultados mostrados, tanto en la Figura 3 como en la Figura 4, se utilizó un α=8, β=1, μ=1, enfocando la eficiencia del algoritmo en la correcta identificación de las notas.

En la Figura 5, se puede ver la diferencia obtenida tanto para el para el MHPS, como para el postprocesamiento, en comparación con el MIDI. Se ve claramente como el algoritmo propuesto, ayuda a eliminar gran cantidad de los falsos positivos generados por los dos algoritmos previos.

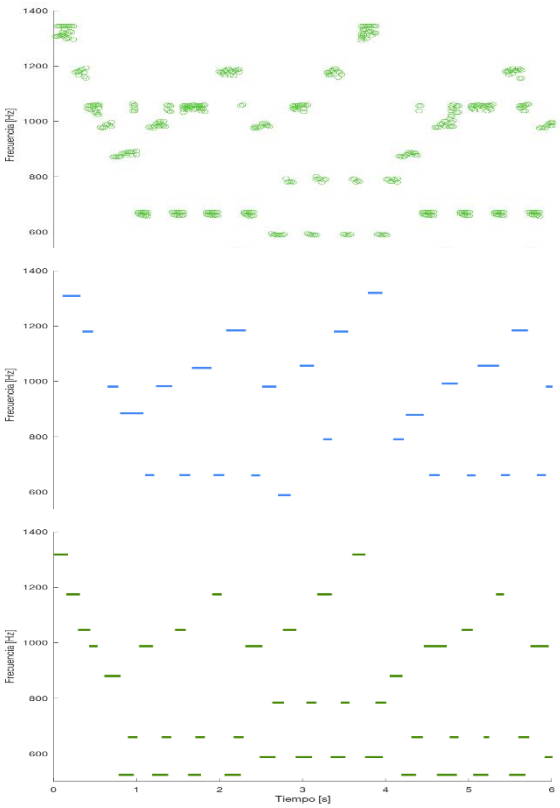

Fig. 5. Resultados MHPS (arriba), propuesta (medio), MIDI (abajo).

V. Conclusiones

Se observó la complejidad que presenta el problema de identificación de notas musicales, al requerir simultáneamente una buena definición temporal como frecuencial. Se vio como la implementación propuesta ayuda considerablemente en la resolución del problema, aumentando la eficiencia de algoritmos conocidos como el HPS y el MHPS, donde la agrupación de notas similares es uno de los principales responsables de este resultado. Y cómo una correcta utilización de la redundancia y de la frecuencia mínima, afectará considerablemente a los resultados obtenidos por el algoritmo. Algo a considerar como futura ampliación de la propuesta, es la inclusión de la potencia de cada nota identificada. En el actual algoritmo esto no se tiene en cuenta, pero es de vital importancia para que el audio generado como salida posea una mayor similitud con el audio original.

Referencias

[1] Pablo Kogan y Jorge P. Arenas, “Eficiencia de la Ponderación “A” desde el Punto de Vista de la Salud”, Universidad Austral de Chile, 2004.

[2] Xuemei Chen y Ruolun Liu, “Multiple Pitch Estimation Based on Modified Harmonic Product Spectrum”, en *International Conference on Information Technology and Software Engineering*, vol. 30, pp. 271-279, 2012.

[3] Jagriti Saini y Rajesh Mehra, “Power Spectral Density Analysis of Speech Signal using Window Techniques”, en *International Journal of Computer Applications*, vol. 131, pp. 33-36, 2015.