Indice.

1. Introducción.....pág.5

2. Objetivos.....pág.7

3. Estado de la cuestión.....pág.8

3.1. Definición de onset.

3.2. Esquema general de un algoritmo de detección de onsets.

3.2.1. Preprocesado.

3.2.2. Obtención de una función de detección.

3.2.2.1. Reducción basada en las características de las señales.

3.2.2.2. Reducción basada en modelos probabilísticos.

3.2.3. Detección de picos.

3.2.3.1. Post-procesado.

3.2.3.2. Umbralización.

3.2.3.3. Proceso de decisión final.

3.2.4. Elección de procesos.

4. Sistemas de detección de onsets.....pág.18

4.1 Reconocimiento de Onsets en Señales de Audio Digital Usando Bancos de Filtros de Un Doceavo de Octava .

4.1.1 Descripción del sistema.

4.1.2 Recepción y preparación de datos.

4.1.3 Desarrollo de una función de detección.

4.1.3.1 Detección de onsets básica.

4.1.3.2 Detección de onset con contexto.

4.1.4 Detección de picos.

4.2 Detección de Onsets en Señales Musicales en el Dominio Complejo.

4.2.1 Anteriores aproximaciones a la detección de onsets.

4.2.1.1 Detección de Onsets basada en la Energía.

4.2.1.2 Detección de Onsets basada en la fase.

4.2.1.3 Aproximación combinada de energía y fase.

4.2.2 Detección de Onsets en el dominio complejo.

4.2.3 Detección de Picos.

4.2.4 Resultados.

4.2.5 Conclusiones.

4.3 Normalización Adaptativa para Mejora de la Detección de Onsets de Audio en Tiempo Real.

4.3.1 Detectores de Onsets actuales.

4.3.2 Selección de Onsets.

4.3.3 Elección de un detector de Onsets.

4.3.4 Problemas con los detectores de onsets.

4.3.5 Normalización Adaptativa.

4.3.5.1 Evaluación.

4.3.5.2 Discusión.

4.3.5.3 Conclusiones.

4.4 Separación de fuentes de Kits de Batería usando una Función de Detección de Percusión y Modulación Espectral.

4.4.1 Metodología.

4.4.2 Estimación temporal.

4.4.3 Modulación espectral.

4.4.4 Resultados.

4.4.5 Conclusiones.

4.5 Segmentación Temporal en Anotación de Música y Audio en Aplicaciones Interactivas.

4.5.1 Modelos perceptuales para segmentación temporal. Funciones de detección de onsets Fase-Vocoder.

4.5.1.1 Contenido en Alta Frecuencia.

4.5.1.2 Diferencia Espectral.

4.5.1.3 Desviación de fase.

4.5.1.4 Distancia en el Dominio Complejo.

4.5.1.5 Distancia Kullback-Liebler.

4.5.2 Perfiles de funciones de detección de onsets.

4.5.3 Selección temporal de picos de onsets de notas.

4.5.4 Postprocesado.

4.5.5 Umbralización dinámica.

4.5.6 Selección de picos en tiempo real.

4.5.7 Puerta de silencio y pre-enmascarado.

4.5.8 Implementación del sistema.

4.5.9 Resultados.

5. Metodología y herramientas para la evaluación de un sistema de detección automática de onsets en música..... pág.90

5.1. Herramientas.

5.2 Etiquetado manual de onsets en señales musicales.

5.2.1. Caracterización de un onset. Particularidades de los onsets en señales musicales.

5.2.2. Metodología de etiquetado manual de onsets.

5.2.3. Elaboración de una base de datos para evaluar el sistema.

5.3. Evaluación de sistemas de detección de onsets.

5.3.1. Sonido sintetizado en los instantes de onsets detectados.

5.3.3 Metodología de evaluación automática.

5.3.2. Inspección de la forma de onda con marcas visuales en los instantes de onsets detectados.

5.3.3. Metodología de evaluación automática.

5.3.3.1. Evaluación automática del sistema.

6. Evaluación de los sistemas en estudio.....pág.101

6.1. Evaluación y resultados.

7. Sistema de Propagación de Interacciones.....pág.108

7.1. Introducción.

7.2. Metodología.

7.2.1. Algoritmo de propagación de corrección de errores para un sistema automático de detección de onsets.

7.2.3 Código.

7.3 Resultados.

7.4 Conclusiones.

8. Conclusiones y trabajo futuro.....pág.123

9. Bibliografía y fuentes de información.....pág.124

1. Introducción.

La música es un fenómeno físico basado en una consecución de eventos, tales como una sucesión de notas con un determinado ritmo, por medio de los cuales un compositor dota de sentido musical a una pieza. Es la forma en la que estos eventos varían en el tiempo la que da el significado a la composición, y hace que un oyente pueda asimilarla y reconocerla.

La segmentación temporal de una secuencia de audio en elementos más pequeños es un paso fundamental en la transformación de sonidos en objetos semánticos. Se han dedicado muchas investigaciones a esta operación, y en las dos últimas décadas , se han desarrollado diferentes algoritmos para separar automáticamente señales musicales en los límites de sus objetos de audio : donde comienza la nota, Onsets, y donde acaba, Offsets [Moelants y Rampazzo, 1997, Klapuri, 1999b]. La extracción de tiempos de inicio es útil en aplicaciones de procesamiento para el modelado preciso de ataques de sonido [Masri, 1996, Jaillet y Rodet, 2001] , ayuda a los sistemas de transcripción a localizar el comienzo de las notas [Bello, 2003, Klapuri, 2004] , y se puede utilizar en los editores de sonido por software para descomponer los archivos de sonido en partes lógicas [Smith , 1996]. Los métodos de detección Onset se han utilizado para la clasificación de música [Gouyon y Dixon, 2004] y la caracterización de patrones rítmicos [Dixon et al., 2004]. Varios sistemas para el seguimiento de tempo hacen uso de la detección de onsets para inferir la localización de latidos [Scheirer, 1998b, Davies y Plumbley, 2004]. Un sistema capaz de detectar estos tiempos de inicio a medida que ocurren , al igual que el oyente humano, permite nuevas interacciones entre instrumentos acústicos y sintéticos [Puckette et al., 1998]. El establecimiento de métodos robustos para la detección de onsets en tiempo real, resulta una tarea importante para la elaboración de instalaciones musicales y sistemas interactivos .

La dificultad de construir un método de detección único que pueda etiquetar todas las observaciones pertinentes es una cuestión abierta. Entre otras cuestiones, en este trabajo estudiaremos varios enfoques para la detección de onsets en audio musical, desde técnicas temporales a bancos de filtros y métodos estadísticos. Estos métodos generalmente se pueden separar en dos tareas : la construcción de una función de detección (ODF) para caracterizar los cambios en la señal, y la selección de picos de esta función , para extraer tiempos de inicio perceptivamente relevantes [Bello et al, 2005.]. Veremos las consideraciones a tener en cuenta para requisitos de tiempo real y como reducir al mínimo el retardo y lograr precisión temporal , dos restricciones requeridas para aproximarse a la capacidad de respuesta del oído humano .

Debido a que la percepción de inicios de notas es un proceso subjetivo del sistema auditivo humano , la evaluación de los métodos de detección de comienzo es una tarea compleja. Se establecerá un marco de trabajo para comparar tiempos de inicio anotados a mano con los obtenidos por diferentes métodos de detección automática. Se evaluarán la localización y precisión de los tiempos de inicio extraídos comparados con las anotaciones manuales , y los costes computacionales de los diferentes métodos.

Además de los sistemas que estudiaremos, en el presente proyecto partimos de un sistema propio capaz de transcribir audio a midi de forma aproximada, cuyo subsistema principal es un detector de onsets que usa un banco de filtros de un doceavo de octava. Se trata de un sistema interactivo el cual ofrece una detección, a priori con errores y en el cual la interacción con el usuario es necesaria. Para corregir los errores el sistema puede, si se desea, propagar la corrección a través de toda la detección para así poder eliminar errores similares. En este trabajo se estudiará la funcionalidad e idoneidad de dicho sistema de propagación.

2. Objetivos.

El objetivo de este trabajo Fin de Grado es estudiar y evaluar diferentes sistemas de detección de onsets, así como, evaluar las interacciones que un usuario experto debería realizar para conseguir la detección esperada. Para ello es necesario crear una base de datos de sonidos etiquetados manualmente con los eventos de inicio de las notas que contiene, así como de un banco de pruebas y análisis de resultados. Con estos datos recopilados se espera poder estudiar la calidad de los algoritmos de detección de onsets y del sistema de propagación de corrección de errores.

Los objetivos concretos de este proyecto son los siguientes:

- Estudiar el estado de la cuestión y la problemática asociada a la detección automática de onsets. Estudiar y analizar modelos existentes de detección de onsets.
- Crear una base de datos de muestras de audio correctamente etiquetada de forma manual con los eventos de inicio de las notas que contiene (Background).
- Estudiar y evaluar diferentes sistemas de detección de onsets a partir de un banco de pruebas con la base de datos creada .
- Evaluar el sistema de propagación de interacciones con el usuario.

3. Estado de la cuestión.

La detección y localización de onsets es un proceso útil para un gran número de técnicas de análisis y catalogación de señales musicales. La manera mas usual de detectar onsets es el estudio de los estados transitorios en las señales: un incremento brusco de energía, un cambio en el espectro de tiempo corto o en las propiedades estadísticas de la señal, etc. Se presenta en este punto, un estudio de la metodología basado en algunas características explícitas y predefinidas de las señales: envolvente de amplitud, magnitud espectral y fase, representación tiempo-frecuencia, modelos probabilísticos, etc. A continuación se presenta un estudio de las técnicas más recurridas a la hora de diseñar un sistema de detección automática de onsets , describiendo las ventajas e inconvenientes de cada método, los problemas que se presentan y las soluciones adoptadas.

3.1. Definición de Onset.

Un onset es el momento en el que comienza un evento musical dentro de una señal de audio. El concepto de onset es muy ambiguo por lo que cada autor hace una interpretación diferente del momento en el que se debe marcar. Para explicarlo con claridad definiremos tres conceptos directamente relacionados, que posteriormente se explicarán en un gráfico que representa el esquema de una nota simple. Estos tres conceptos son: ataque, transitorio, onset.

- Ataque: Para una nota es el intervalo de tiempo en el que la envolvente de amplitud incrementa.
- Transitorio: Es el intervalo de tiempo en el que la envolvente de la señal cambia rápidamente y de una manera impredecible. Este concepto esta directamente relacionado con la resolución temporal en el análisis de la señal, ya que, se asume que el oído humano no puede distinguir entre dos transitorios separados menos de 10ms.
- **Onset**: Es un único instante de tiempo discreto elegido para marcar el origen de un transitorio.

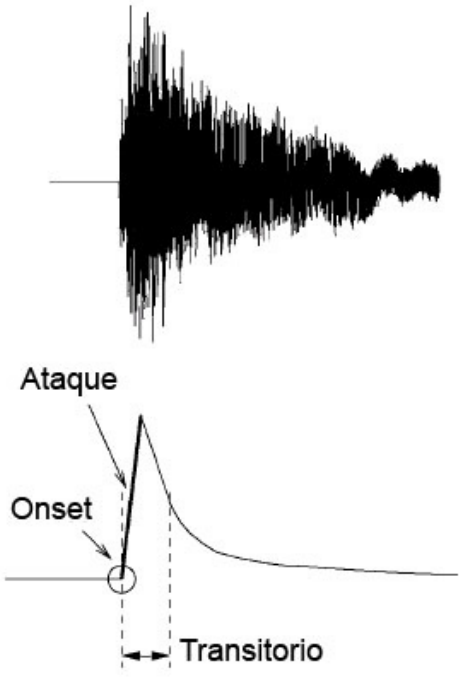


Fig. 2.1. Concepto de Ataque, Transitorio y Onset en el caso ideal de una nota simple.

3.2. Esquema general de un algoritmo de detección de onsets.

La figura 2.1 muestra el caso ideal de un onset aislado. En la mayoría de casos reales, las señales no ofrecen este aspecto, si no que se nos presentan como señales polifónicas, multitímbricas, o con presencia de ruido, que hacen que coincidan varios transitorios en el mismo instante de tiempo o parcialmente solapados, lo que hace casi imposible su detección directamente sobre la señal de audio original en el dominio del tiempo. Por este motivo es necesario algún tipo de procesamiento de la señal para segmentarla adaptarla y simplificar la estructura local de la señal original, realzando los elementos necesarios para facilitar la detección de onsets, y atenuando los factores que dificultan dicha labor. Este proceso ofrece múltiples opciones, y su elección depende de factores como la propia naturaleza de la señal, aplicación a la que se dirige, necesidad de procesado a tiempo real, recursos de computación, etc.

La mayoría de algoritmos de detección de onsets hacen un "preprocesado" para adaptar la señal a sus necesidades, extrayendo de ésta los elementos necesarios para construir una nueva señal con información relevante de la presencia de eventos discretos. Este proceso da lugar a lo que en la literatura se conoce como "función de detección", sobre la que se suele aplicar un "algoritmo de detección de picos" que finalmente proporcione la información deseada.

El esquema empleado en la mayoría de algoritmos de detección de onsets consta de tres grandes bloques:

- Preprocesado.
- Obtención de una Función de Detección ODF (adaptación, reducción).
- Detección de Picos e interpretación.

La siguiente figura representa el diagrama de bloques de un detector de onsets típico:

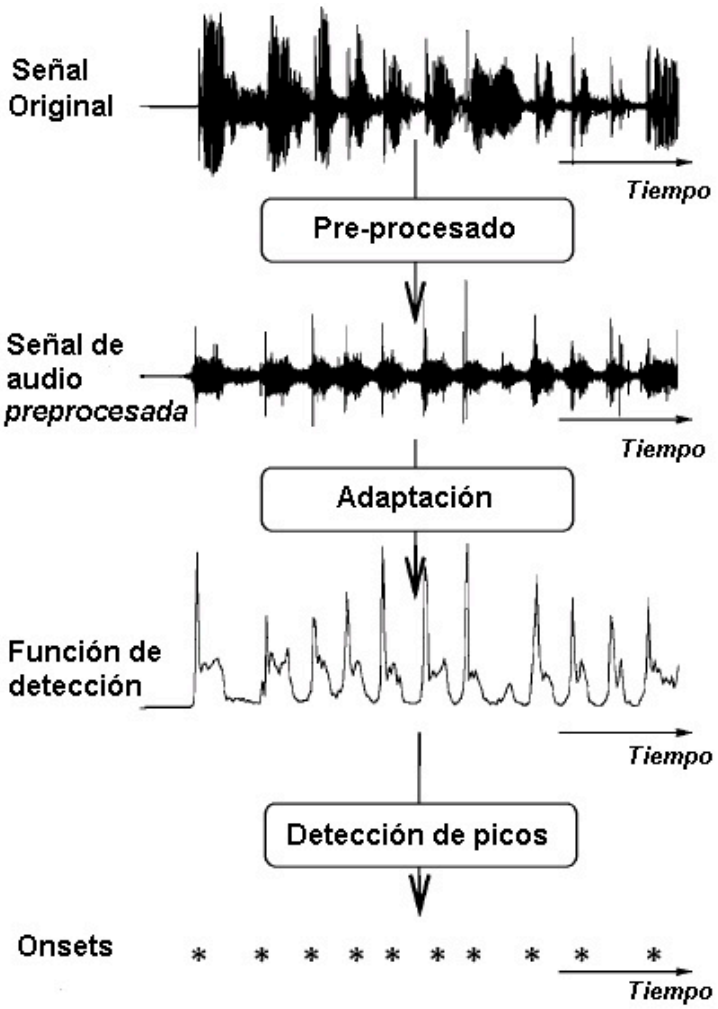


Fig. 2.2. Esquema genérico de un detector de onsets.

3.2.1. Preprocesado.

El concepto de preprocesado implica la transformación de la señal original con el fin de acentuar o atenuar varios aspectos de esta, de acuerdo a la relevancia que tienen en la cuestión a tratar. Es un paso opcional, pero generalmente adoptado y de gran importancia para los procesos que se desarrollan a continuación.

Hay infinidad de tratamientos que pueden aplicarse a una señal musical a fin de facilitar la detección de onsets. Sin embargo, casi todos los algoritmos recurren a dos técnicas: separar la señal en múltiples bandas de frecuencia, o separar entre estados transitorios y estados estables. Los métodos de división en múltiples bandas de frecuencia se utilizan cuando la aplicación requiere una detección individual por sub-bandas para la detección global, o también, para dotar de una mayor robustez al sistema. Normalmente los bancos de filtros se diseñan a partir de simulaciones de percepción del oído y de procesos psicoacústicos. Por otro lado la separación entre estados transitorios está normalmente asociada al modelado de señales musicales, pero algunos métodos como la producción de señales modificadas (residuales, señales transitorias,…) son usados en algoritmos de detección, ofreciendo buenos resultados.

3.2.2. Obtención de una función de detección.

En el contexto de la detección de Onsets, a este proceso se le suele llamar “Reducción”, refiriéndose al proceso de transformación de una señal de audio en otra que pone de manifiesto los sucesos transitorios de la señal original. Este proceso es la clave de la detección. Los métodos de reducción se dividen, básicamente en dos grandes grupos:

- Métodos basados en el uso de las características de señales definidas explícitamente.
- Métodos basados en modelos de señales probabilísticas.

3.2.2.1. Reducción basada en las características de las señales.

Características temporales: Al observar la evolución temporal de una señal musical simple, es notable que la aparición de un onset esta normalmente acompañada de un incremento en la amplitud de la señal. Los primeros métodos de detección de onsets utilizaban una función de detección que seguía la amplitud de la envolvente de la señal. Un seguidor de envolvente puede ser fácilmente implementado por medio del rectificado y alisado de la señal mediante, por ejemplo, un filtrado paso-bajo. Este método ofrece resultados satisfactorios para algunas aplicaciones donde hay transitorios percusivos fuertes que se presentan sobre un fondo silencioso. Debido al alisamiento de la señal, no se puede usar un detector de picos para la detección, por ello una alternativa es trabajar con derivadas en el tiempo de la energía, de manera que un crecimiento súbito de esta, se transforma en picos estrechos y notables, que facilitan el trabajo del detector de picos. Estos métodos son normalmente usados en combinación con los métodos de preprocesado descritos anteriormente.

Otra vertiente se basa en los principios de la psicoacústica, considerando la percepción logarítmica en la audición. Calculando el logaritmo de la derivada y combinando esto con el método de la separación en múltiples bandas de frecuencia, conseguimos simular la respuesta auditiva del oído, resultando una notable reducción en la aparición de onsets falsos respecto a los métodos de modulación de amplitud.

Características espectrales: Un gran número de algoritmos de detección emplean la estructura espectral de la señal para producir una función de detección más fiable, a la vez que reducen la necesidad de preprocesado y se introduce la detección en señales polifónicas con múltiples instrumentos. Se pueden considerar varias formas de transformada, tamaños y tipos de ventana, y demás parámetros para conseguir la representación espectral de la señal. Después del proceso de transformada se puede recurrir a una ponderación en función de la frecuencia, ya que la energía de las señales suele concentrarse en las bajas, medias-bajas, frecuencias. Se suele generar un banco de bandas de frecuencia que se analizan independientemente, calculando distancias entre cada una, para luego combinarlas en la detección final, resultando así una detección mucho más robusta. Otro sistema consiste en correlar las distintas distancias entre bandas con tablas de

señales predefinidas, cuyos máximos muestran los picos que en el tiempo se corresponden con la función de detección buscada.

Características espectrales usando la fase: Todos los métodos descritos anteriormente tiene en común el uso de la magnitud del espectro como única fuente de información. Sin embargo algunos algoritmos hacen también uso de la información que ofrece la fase del espectro para analizar la posición de onsets. Esta técnica se usa tanto independientemente, como para complementar a otras. Se extrae la fase calculada a partir de la transformada para determinar los valores de frecuencia instantánea y una estimación de la frecuencia actual del componente de la transformada dentro de su ventana. La desviación de fase se define como la segunda derivada de la fase. Estudiando estos dos parámetros y sabiendo que, para una sinusoide localmente estacionaria estos parámetros deberían ser aproximadamente constantes respecto a las ventanas adyacentes, se pueden sacar conclusiones de eventos en la señal.

Análisis en Tiempo-Frecuencia (TRF) y Tiempo-escala: Una alternativa al estudio de la envolvente temporal de la señal, y los coeficientes espectrales, es el uso de representaciones en tiempo-escala y tiempo-frecuencia. Para ello se utilizan técnicas como medir la disimilitud entre las características de vectores correspondientes a TRF de clases de Cohen discretizadas, y el resultado de convolucionar una TRF Wigner-Ville con una función de núcleo Gaussiano. Este método de apariencia tan laboriosa es parecido a la aproximación de diferencias espectrales, y eligiendo un núcleo apropiado la representación puede llegar a ser equivalente al espectrograma de la señal. Otra aproximación para la detección de transitorios esta basada en una descomposición Wavelet simple de una señal residual. Se usa esta transformada (Haar Wavelet) por su simplicidad y su buena localización temporal en escalas pequeñas. Este esquema se basa en la correlación cruzada entre escalas de coeficientes: coeficientes wavelet grandes están relacionados con transitorios en la señal, y si un coeficiente tiene una amplitud grande, existe una alta probabilidad de que los coeficientes con la misma localización temporal a escalas mas pequeñas también tengan una gran amplitud, y de esta forma construir árboles de coeficientes significativos. Cuando encontramos una rama del tamaño completo, de la escala mas grande a la mas pequeña, se puede cuantificar como un “modulo de

regularidad" el cual nos da una medida de la regularidad local de una señal. Estos módulos de regularidad pueden actuar como una función de detección de onsets.

3.2.2.2. Reducción basada en modelos probabilísticos.

- **Métodos de Modelos Basados en Cambios del Punto de detección**: Esta aproximación esta basada en el test de "proporción de probabilidad secuencial". Este presupone que las muestras de una señal están generadas de uno de dos modelos estadísticos "a" o "b". Se define el logaritmo de grado de probabilidad como el cociente de la función de densidad de probabilidad del modelo "b" entre "a". Según el modelo bajo el que se esté actuando en el momento tendremos una función de expectación diferente. Si se asume que en un principio que la señal sigue el proceso "a" y al cabo de un tiempo, a priori desconocido cambia a "b", entonces también cambiará el logaritmo de grado de probabilidad de la señal, y por ende la señal. Este logaritmo de grado de probabilidad es el que se usa como función de detección. Cuando los modelos son desconocidos se estiman a partir de los datos que aporta la propia señal.

- **Aproximaciones basadas en "Señales Sorpresa"**: El método descrito anteriormente busca cambios instantáneos entre dos modelos distintos. Una alternativa consiste en buscar "momentos sorpresivos" relativos a un sólo modelo global. Para este fin, la función de detección esta definida a pequeños trazos de tiempo del negativo del logaritmo de la probabilidad de la historia reciente de la señal dada, de acuerdo al modelo global.

3.2.3. Detección de picos.

Si la función de detección ha sido convenientemente diseñada, los cambios bruscos producidos en la señal, tales como onsets, se podrán identificar de manera óptima a partir de las características de dicha función. De forma general, las características que representan onsets dentro de la función de detección son máximos locales, generalmente

sujetos a niveles de variabilidad en tamaño y forma, y enmascarados por ruido, que puede ser debido al propio ruido de señal, o a aspectos de esta que no son específicamente onsets, como pueden ser los trémolos o vibratos en algunas notas. Por este motivo, para estimar el tiempo de un onset dentro de los eventos producidos en una señal, es necesario un algoritmo de detección de picos robusto. El proceso de detección de picos en una función de detección puede dividirse en tres pasos:

- Post-procesado.
- Umbralización.
- Proceso de decisión final.

3.2.3.1. Post-procesado.

Al igual que el preprocesado el post-procesado es un paso opcional, que depende del método de reducción usado para implementar la función de detección. El propósito del post-procesado es facilitar la tarea de la umbralización y la detección de picos final, incrementando la uniformidad y la consistencia de las características relativas a los eventos en la función de detección, y transformándolos en máximos locales aislados y fácilmente detectables. En esta categoría entran los procesos de alisado, reducción de ruido y todos los necesarios para facilitar la elección de parámetros de umbralización, tales como normalización y eliminación de componente continua.

3.2.3.2. Umbralización.

Para cada tipo de función de detección, e incluso después del post-procesado, quedaran una gran cantidad de picos que realmente no son onsets. Por este motivo es necesario definir un umbral que sea efectivo a la hora de clasificar los eventos que sean onsets y los que no. Existen dos aproximaciones para establecer este umbral:

- Umbralización fija.
- Umbralización adaptativa.

Umbralización fija: Los métodos de umbralización fija definen los onsets como picos de la función de detección que exceden un umbral. Esta aproximación puede funcionar bien con señales de pequeño rango dinámico, aunque en la música es normal encontrarse con significativos cambios de nivel a lo largo de una pieza. En estas situaciones, en las que tenemos un gran rango dinámico, un umbral fijo tendería a pasar por alto onset en pasajes de bajo nivel, y a sobredetectar onsets en los pasajes con mayor nivel. Por esta razón, en algunos casos se requiere un método de umbralización adaptativa.

Umbralización adaptativa: Generalmente un umbral dinámico se calcula como la versión alisada de la función de detección. Este alisamiento puede ser lineal, por ejemplo utilizando un filtrado paso bajo FIR, o por el contrario no lineal usando, por ejemplo, el cuadrado de la función de detección. Sin embargo, un umbral calculado de esta forma puede presentar grandes fluctuaciones cuando en la función de detección hay un pico de gran valor, tendiendo a enmascarar los onsets de menor valor adyacentes a éste.

3.2.3.3. Proceso de decisión final.

Tras el post-procesado y la umbralización de la función de detección, el proceso de detección de picos se reduce a identificar los máximos locales que superen el umbral definido.

3.2.4. Elección de procesos.

La elección de los métodos a aplicar viene dada por varios factores como el tipo de señal a analizar, recursos de computación, necesidad de procesado a tiempo real, etc. Por regla general se escoge aquel sistema que satisface los requisitos de la aplicación y que tiene una menor complejidad. Para ello hay que llegar a un equilibrio de complejidad entre preprocesado, función de detección y detección de picos.

4. Sistemas de detección de onsets.

Como ya se ha comentado, existen numerosos estudios y enfoques sobre detección de onsets. Uno de los objetivos de este trabajo fin de grado es estudiar algunas de estas aproximaciones, basadas en distintas técnicas y modelos de los descritos en el apartado anterior. Otro de los objetivos de este trabajo es analizar y evaluar el funcionamiento de los sistemas en estudio a partir de un banco de pruebas de señales de audio con los tiempos de onsets etiquetados manualmente. Por ello se ha seleccionado un conjunto de algoritmos de detección de onsets que disponen de un plug-in Vamp para el software de estudio de señales de audio Sonic Visualizer, sobre el cual se hablará en el apartado de herramientas utilizadas. De este modo, se tendrá acceso a 3 sistemas diferentes que permiten elegir distintos algoritmos para construir la función de detección. Dichos sistemas son:

- **"Recognition of Note Onsets in Digital Music Using Semitone Bands"**. Antonio Pertusa, Anssi Klapuri y Jose Manuel Iñesta. Departamento de Lenguajes y Sistemas Informáticos , Universidad de Alicante, España. Signal Processing Laboratory, Tampere University of Technology, Finland.

- **"Complex Domain Onset Detection for Musical Signal"**. Chris Duxbury, Juan Pablo Bello, Mike Davies and Mark Sandler. Department of Electronic Engineering Queen Mary, University of London Mile End Road, London, UK

- **"Adaptative Whitening for Improved Real-Time Audio Onset Detection"**. Dan Stowell, Mark Plumbey. Centre for Digital Music Queen Mary, University of London.

- **“Drum Source Separation using Percussive Feature Detection and Spectral Modulation”**. Dan Barry, Derry Fitzgerald, Eugene Coyle and Bob Lawlor. Digital Audio Research Group, Dublin Institute of Technology, Ireland, Dept. Electrical Engineering, Cork Institute of Technology, Cork, Ireland, Dept. of Electronic Engineering, National University of Ireland, Ireland.

- Aubio Onset Detection Library by Paul Brossier. Chapter 2 from Paul M. Brossier´s Thesis , **“Automatic Annotation of Musical Audio for Interactive Applications”**, under the direction of Dr. Mark Plumbley, Prof. Mark Sandler, Prof. Eduardo R. Miranda y Dr. Michael Casey, Centre for Digital Music Queen Mary University of London. UK.

4.1 Reconocimiento de Onsets en Señales de Audio Digital Usando Bancos de Filtros de Un Doceavo de Octava .

Se presenta un sistema de detección automática de *onsets* en el que se propone un algoritmo de análisis de la señal de audio, basado en el filtrado de la misma por medio de un banco de filtros de 1/12 de octava. La salida del filtro será analizada teniendo en cuenta uno o varios instantes de tiempo, dando lugar a dos metodologías diferentes: detección básica y detección con contexto. En el sistema básico de detección, a la salida del filtro se evalúa la evolución temporal de la energía en cada banda a través la derivada temporal de cada una de las bandas en el instante actual y el anterior. En el sistema de detección con contexto, la evolución temporal de la energía se evalúa considerando un mayor número de instantes de tiempo, según las características de la señal analizada. Por último el algoritmo describe un proceso de decisión en el que se establecen los tiempos de *onsets*, estudiando la dinámica de la energía presente para cada instante de la señal.

El objetivo de este sistema es conseguir un sistema de detección en tiempo real, que sea sensible a los *onsets* en la música, pero robusto frente al ruido y las variaciones en el tiempo del espectro que no representen un *onset* de una nota musical. Este trabajo aborda la detección de instantes de comienzo de notas musicales en señales de audio digital, por medio de un sistema que analiza la información del espectro a través de un banco de filtros

de 1/12 de octava, para después computar sus diferencias relativas en el tiempo y obtener así una función de detección. Finalmente los picos de dicha función que superan un umbral son considerados como *onsets*.

Los instrumentos en la música occidental, están afinados siguiendo la escala Bien Temperada, formada por las siete notas de la escala cromática y sus correspondientes sostenidos o bemoles, dando lugar a doce notas diferentes [*do, do#, re, re#, mi, fa, fa#, sol, sol#, la, la#, si*] separadas entre sí un semitono. Esta escala se repite cuando la primera de las notas dobla su frecuencia fundamental, volviendo a recibir el mismo nombre. De esta forma, los sonidos de la música occidental se organizan en octavas (distancia entre una nota y otra que dobla su frecuencia, y que recibe el mismo nombre) divididas en doce notas cada una.

Frecuencias (Hz)									
	$Octava_{-1}$	$Octava_0$	$Octava_1$	$Octava_2$	$Octava_3$	$Octava_4$	$Octava_5$	$Octava_6$	$Octava_7$
C		32,7	65,4	130,8	261,6	523,2	1046,4	2092,8	4186
C#		34,64	69,28	138,56	277,12	554,24	1108,48	2216,96	
D		36,7	73,4	146,8	293,6	587,2	1174,4	2348,8	
D#		38,88	77,76	155,52	311,04	622,08	1244,16	2488,32	
E		41,19	82,38	164,76	329,52	659,04	1318,08	2636,16	
F		43,64	87,28	174,56	349,12	698,24	1396,48	2792,96	
F#		46,23	92,46	184,92	369,84	739,68	1479,36	2958,72	
G		48,98	97,96	195,92	391,84	783,68	1567,36	3134,72	
G#		51,89	103,78	207,56	415,12	830,24	1660,48	3320,96	
A	27,5	55	110	220	440	880	1760	3520	
A#	29,14	58,27	116,54	233,08	466,16	932,32	1864,64	3729,28	
B	30,87	61,73	123,46	246,92	493,84	987,68	1975,36	3950,72	

Tabla 3.1. Posibles frecuencias generadas por un piano de concierto.

Un banco de filtros de 1/12 de octava, basado en la formación musical, ofrece ciertas ventajas respecto a otros bancos de filtros usados en otros sistemas, como por ejemplo, el banco de filtros de Mel y bancos de filtros de simulación psico-acústica, que intentan simular la respuesta del oído humano. Cuando una nota comienza a sonar, la mayor parte de la energía se concentra en su frecuencia fundamental y en los armónicos cuya

frecuencia central se sitúa en el centro de las bandas de semitonos. Este es el motivo por el cual este detector de *onsets* utiliza un banco de filtros de 1/12 de octava en el dominio de la frecuencia, donde las frecuencias centrales de cada banda se corresponden con las frecuencias fundamentales de las notas musicales que están dentro del rango del filtro.

Usando bandas de semitonos el efecto de las variaciones sutiles en el espectro producidas durante los estados de relajación y mantenimiento (sustain) de una nota se minimizan. Debido a que mientras que una nota esta sonando dichas variaciones ocurren próximas a las frecuencias centrales de las bandas de 1/12 de octava, el valor de salida de una banda durante el proceso de ataque será similar en el instante siguiente, ya que estas variaciones ocurren dentro de la misma banda, por lo que se consigue evitar la detección de *onsets* "falsos positivos". Por otro lado, cuando comienza una nota producida por un instrumento afinado, el valor de salida de la correspondiente banda incrementa de manera significativa debido a que la mayor parte de la energía de sus armónicos se suele concentrar en las frecuencias centrales de las bandas de semitono. Por este motivo, este sistema es especialmente sensible a las variaciones de frecuencia mayores que un semitono. Además gracias al efecto de enfatizar las variaciones del espectro al comenzar una nota y minimizarlas cuando esta sonando, el sistema presenta una gran robustez frente a técnicas interpretativas como vibratos (no mayores que un semitono), y los *pitch bend* (glissando). Normalmente cuando se presenta un *pitch bend*, un nuevo *onset* no deseado es detectado cuando éste alcanza un cuarto de tono por encima o por debajo de la afinación inicial. Además, estas características hacen de este detector un algoritmo adecuado para desarrollar sobre él un sistema de trascripción musical, puesto que las "unidades de afinación" están medidas en semitonos.

4.1.1 Descripción del sistema.

Para implementar este sistema detector de *onsets*, se desarrolla un algoritmo que consta de tres procesos principales divididos a su vez en varios subprocesos:

- Recepción y preparación de datos. Preprocesado y adaptación para la siguiente etapa.
  - Lectura de fichero de audio digital.
  - Enventanado y solapamiento.
  - Transformación al dominio frecuencial.

- Desarrollo de una función de detección.
  - Aplicación del banco de filtros de 1/12 de octava.
  - Diferencias temporales relativas en cada banda, basadas en sus primeras derivadas según la necesidad de contexto.
  - Normalización de los valores de la función de detección.

- Interpretación de la función de detección, búsqueda de *onsets*.
  - Detección de picos.
  - Umbralización.

A continuación se explican cada uno de estos procesos, sus características y parámetros.

4.1.2 Recepción y preparación de datos.

La entrada al sistema es un archivo de audio digital. A fin de eliminar las componentes de frecuencia no usadas e incrementar a su vez la resolución espectral, se remuestrea la señal pasando a 22,050 Hz. Una vez resampleada la señal, según el teorema de Nyquist, estaremos trabajando con frecuencias de hasta $fs/2 = 11{,}025$ Hz, la cual es lo suficientemente alta como para cubrir el rango de sonidos que suele usarse en las piezas musicales. Además, para evitar los sonidos coincidentes en el panorama esteréo, que se considerarían como un solo *onset*, se han mezclado los dos canales, dejando la entrada en formato monofónico, y ahorrando así la mitad de procesado al sistema. A fin de poder analizar las características frecuenciales de la señal, se aplican las técnicas de enventanado y solapamiento de ventanas, cuyos parámetros, (tipo de ventana, numero de muestras N, y porcentaje de solapamiento O) se establecen definitivamente de manera experimental. Una vez que se establecen estos parámetros la resolución temporal Δt del sistema podrá ser calculada como:

$$\Delta t = \frac{(1-O)N}{fs} \quad \text{(Eq.1)}$$

Tras los citados procesos de enventanado y solapamiento, se aplica una Transformada Rápida de Fourier para conseguir el espectrograma de la señal y poder analizar sus características frecuenciales. En este procesado de la señal es muy importante la caracterización de los parámetros que se van a usar. Como ya se ha indicado anteriormente, la resolución temporal Δt viene determinada por el tamaño de la ventana N, el porcentaje de solapamiento O y la frecuencia de muestreo fs. A partir de esta resolución temporal Δt podemos calcular la resolución frecuencial Δf como:

$$\Delta f = fs/N \quad \text{(Eq.2)}$$

Estos datos establecen el rango de frecuencias que a analizar y la precisión del sistema a la hora de discernir entre distintos valores. En un piano de concierto el rango de frecuencias va desde la nota $G\#_{-1}$ (25.9 Hz) hasta la nota C_7 (4186 Hz). Este detector trabaja con bandas de un semitono, por lo que la capacidad de análisis en la bandas graves dependerá de la resolución espectral del sistema, ya que por ejemplo, para poder analizar la banda que va desde el $G\#_{-1}$ (de frecuencia 25.9 Hz) hasta el A_{-1} (de frecuencia 27.5 Hz), se necesitarís una resolución frecuencial mínima de 1.6 Hz.

4.1.3 Desarrollo de una función de detección.

Tras llevar a cabo la transformación al dominio de la frecuencia de la señal de entrada, nos encontramos con una señal de N componentes espectrales, en cada instante de tiempo del archivo de audio digital a procesar. Este espectro obtenido de la STFT (Short Time Fourier Transform) es repartido a través de un banco de filtros de B bandas para simular la repuesta de un filtro de 1/12 de octava en el dominio de la frecuencia. Estas B bandas estas construidas siguiendo una escala logarítmica dentro del rango de frecuencias permitidas por los valores de los parámetros explicados en el apartado anterior.

Para construir las bandas de 1/12 de octava, se utilizan un conjunto de ventanas triangulares de diferentes tamaños, centradas en cada una de las notas que hay dentro del rango de frecuencias con el que podemos trabajar. Además para no dejar componentes de frecuencia sin evaluar estas ventanas están solapadas, de manera que aunque los eventos que se producen centrados en las frecuencias de las notas musicales tienen un mayor peso a la salida del filtro, los que ocurren mas alejados de estas también se tienen en cuenta. Las ventanas están construidas en base a tres frecuencias: la frecuencia central de la banda y la frecuencias centrales de las dos adyacentes. De esta manera la frecuencia que actúa como límite superior de una banda lo hace a su vez como frecuencia central de la siguiente, siendo esta también el límite inferior de la que vendrá a continuación. Las aristas del triángulo son los valores de ponderación de cada una de las componentes espectrales que contiene la banda que delimita, y que van desde 0 en los extremos a 1 en el centro.

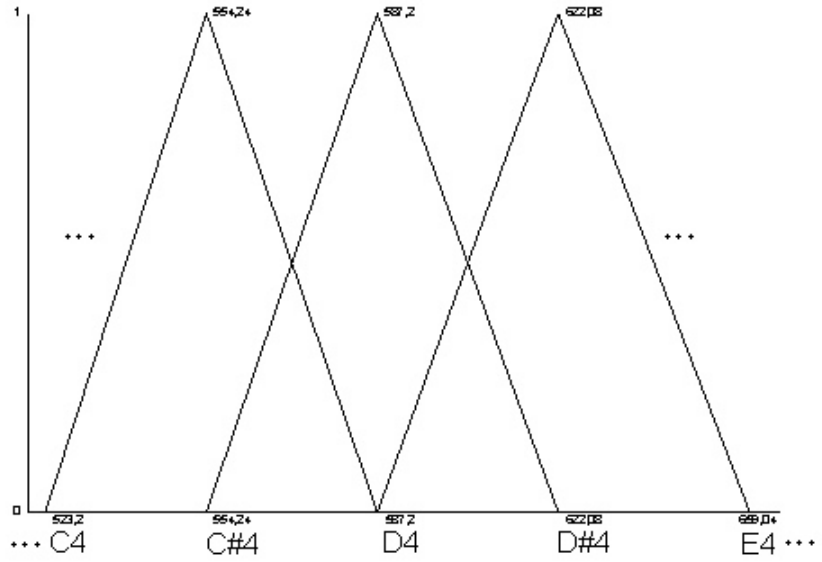


Fig.3.1.- Ejemplo de ventanas triangulares para las notas C4, C#4, D4, D#4, E4.

Al construir las bandas con ventanas de diferentes tamaños, las situadas sobre las componentes espectrales más agudas tendrán un mayor número de elementos, mientras las correspondientes a las bajas frecuencias tendrán a muy pocos. Por lo tanto si la señal de entrada es un ruido uniformemente distribuido, las bandas más anchas tendrán valores de salida más altos que las estrechas. A fin de minimizar este problema, para calcular el valor de salida de cada banda se utiliza el valor RMS (Root Mean Square). Esta medida pondera el valor de cada componente por el valor correspondiente de la ventana triangular, dando así mayor énfasis a los valores más altos del espectro. De esta forma el valor de cada una de las bandas del filtro $b_k(t)$, en el instante t vendrá dada por la ecuación:

$$b_k(t) = \sqrt{\sum_{j=1}^{Wk} (X(j,t)w_{kj})^2} \quad \text{(Eq.3)}$$

Siendo $\{w_{kj}\}_{j=1}^{Wk}$ los valores de la ventana triangular para cada banda, W_k el tamaño de la ventana *k*-ésima y X las muestras del espectro correspondientes a dicha ventana en el tiempo t, con j como índice de la componente de frecuencia.

Se usa el valor RMS de cada de cada banda en lugar de la energía debido a que, usando la energía, las pequeñas variaciones en bandas de mayor anchura se enfatizan, dando lugar a falsos *onsets* durante el estado de mantenimiento de algunas notas. Por otra parte se evita que algunos *onsets* suaves queden enmascarados por otros más fuertes.

En este punto del sistema, nos encontramos con un conjunto de B valores correspondientes a cada k-ésima banda de nuestro banco de filtros. A fin de ver como evoluciona la presencia de energía o los cambios de esta en cada nota musical estos valores serán calculados de forma independiente en cada banda, para varios instantes de tiempo según la necesidad de evaluar un contexto.

De esta forma encontramos dos casos de evaluación para cada banda:

- Reconocimiento básico de *onset*.
- Reconocimiento de *onset* con contexto.

4.1.3.1 Detección de onsets básica.

En este caso se estudia la evolución espectral de la señal filtrada mediante la primera derivada temporal en cada banda. En este caso la derivada en cada banda es calculada de la siguiente manera:

$$c_k(t) = \frac{d}{dt} b_k(t) \quad \text{(Eq.4)}$$

Una vez obtenida la derivada, para cada banda, entre el instante de tiempo actual y el anterior, se combinan para extraer un único valor general para ese instante de tiempo. Para ello se hace la suma del valor de la derivada de primer orden en cada una de las bandas para cada instante de tiempo. A fin de detectar solo el comienzo de las notas, en la suma solo se acumulan los valores correspondientes a las derivadas positivas, es decir, aquellos que indican un incremento de energía en ese instante. Los valores negativos no se

consideran. La ecuación de este proceso para cada instante de tiempo queda de la siguiente forma:

$$a(t) = \sum_{k=1}^{B} \max\{0, c_k(t)\} \quad \text{(Eq.5)}$$

Para normalizar la función de detección en el rango [0,1] se introduce un último paso en su construcción. Se calcula el sumatorio $s(t)$ de los valores de todas las bandas para cada instante de tiempo.

$$s(t) = \sum_{k=1}^{B} b_k(t). \quad \text{(Eq.6)}$$

A $s(t)$ se le aplica un umbral de silencio de manera que si $s(t) < \mu$, entonces la función de detección será $o(t) = 0$ en ese instante. Esto es así, debido a que un instante de tiempo que no tenga energía, o esta sea de una amplitud muy pequeña, probablemente no sea un *onset*, por lo que ahorramos tiempo de procesado, y evitamos la aparición de falsos positivos.
Finalmente se divide la suma de las derivadas positivas entre la suma de las amplitudes de las bandas, para obtener una diferencia relativa. Después de este proceso la función de detección $o(t)$ pertenece al rango [0,1], $o(t) \in [0,1]$.

$$o(t) = \frac{a(t)}{s(t)} \quad \text{(Eq.7)}$$

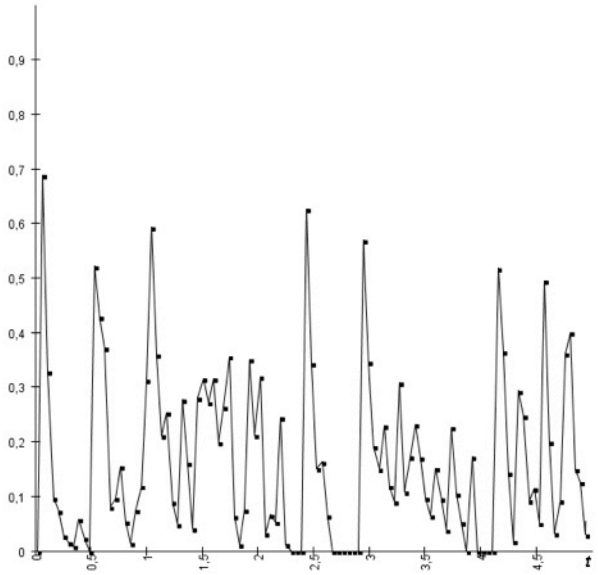


Fig.3.2.- Función de detección.

4.1.3.2 Detección de onset con contexto.

La metodología descrita anteriormente presenta buenos resultados para instrumentos como el piano o la guitarra, los cuales tienen una forma muy definida en la envolvente de ataque. Existen otro tipo de instrumentos que tienen tiempos de ataque más largos, tales como los órganos de iglesia, o que presentan movimientos en sus armónicos durante la fase de sostenimiento, como algunos instrumentos de cuerda o guitarras eléctricas distorsionadas. Para este tipo de instrumentos en los que la envolvente de ataque no es tan concreta, se presenta la necesidad de estudiar la evolución de la energía considerando un mayor número de ventanas, es decir usando un contexto. La metodología en este caso es similar a la aplicada para el reconocimiento básico pero reemplazando la ecuación 4 por esta otra, en donde la variable *C* (contexto) representa el número de ventanas a considerar:

$$\hat{c}_k(t) = \sum_{i=1}^{C} i \times [b_k(t+i) - b_k(t-i)] \quad \text{(Eq.8)}$$

Esta ecuación es una variación de otra usada para mejorar el rendimiento en algunos sistemas de reconocimiento del habla. Cada frame que interviene en esta operación esta ponderado. Para C=1 se usan los frames anterior y posterior para llevar a cabo el cálculo de

la diferencia. En el caso C=2 el periodo de tiempo considerado incluye además de los instantes adyacentes al frame actual, los dos que delimitan a éstos. Es decir, entran en juego 5 instantes de tiempo, siendo el central el instante actual, los dos de los extremos operandos de la resta de mayor ponderación, y los dos interiores los operandos de la otra diferencia. En la fórmula, el papel del factor de ponderación está representado por la variable i.

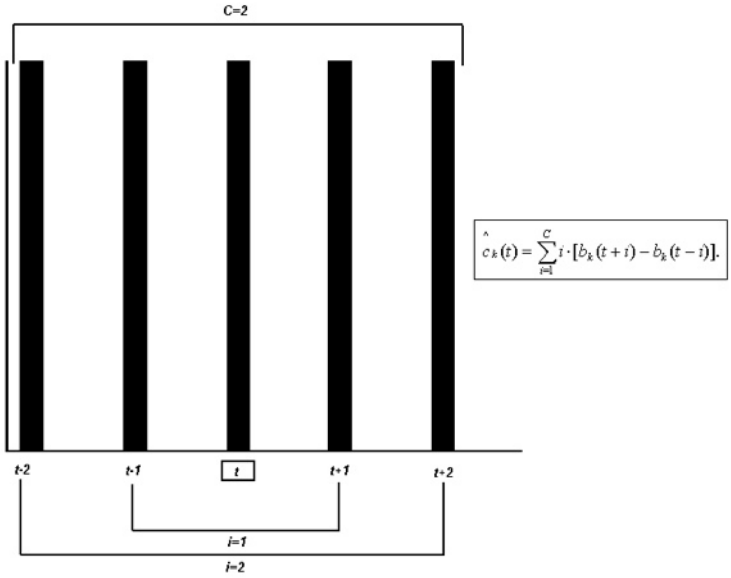


Fig.3.3.- Instantes de tiempo evaluados según el contexto.

De la misma manera que en el caso para detección básica, se normaliza la función de detección reemplazando la ecuación 6 por la siguiente, en la que además de la energía local se considera la de los instantes adyacentes, ponderada según el contexto:

$$\hat{s}(t) = \sum_{k=1}^{B} \sum_{i=1}^{C} i \times b_k(t+i) \quad \text{(Eq.9)}$$

De esta manera $\hat{s}(t)$ representa el valor máximo que puede tener $a(t)$ usando un contexto. Al igual que en el caso para detección básica a $s(t)$ se le aplica un umbral de silencio de manera que si $s(t) < \mu$, entonces la función de detección será $o(t) = 0$ en ese instante.

4.1.4 Detección de picos.

Una vez obtenida la función de detección $o(t)$, se detectan los picos que están sobre un umbral para determinar la existencia de *onsets* en cada instante de tiempo.

El proceso de detección de picos en la función de detección consiste en procesar los elementos de la misma de tres en tres, y seleccionar el elemento central como posible candidato a *onset*, si su valor es mayor que el de los dos elementos laterales. Este proceso se realiza para cada instante de tiempo.

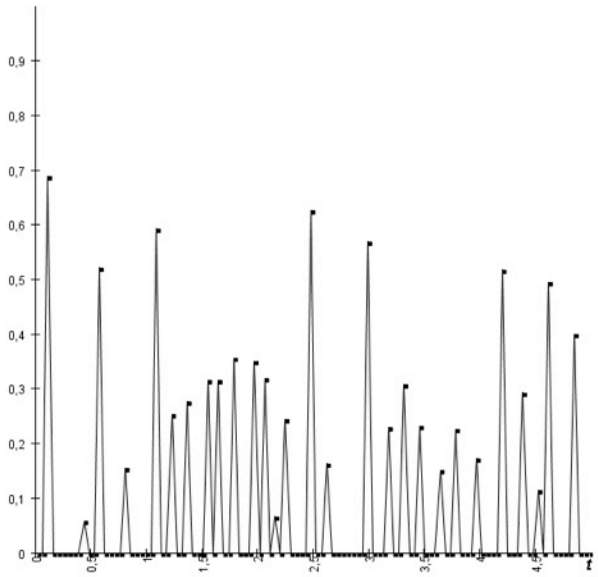


Fig.3.4.- Detección de picos en la función de detección.

Debido al hecho de que sólo los picos de la función de detección son tomados en cuenta como posibles candidatos a *onset*, dos *onset* consecutivos para los instantes t y t+1 no pueden ser detectados, ya que los dos no pueden ser un pico. De esta forma para el caso de detección básico, la diferencia temporal mínima entre dos *onsets* será el doble de la

resolución temporal del sistema ($2 \times \Delta t$) , ya que las diferencias relativas están calculadas en base a un instante de tiempo t y al anterior t-1.

En el caso de detección para instrumentos complejos, la distancia temporal mínima entre *onset* viene determinada por el valor del contexto, siendo mayor cuanto mayor sea *C*. Para que una nota pueda ser detectada de manera fiable dentro de un contexto, la duración *l* de la misma debe cumplir la siguiente desigualdad:

$$l \geq \Delta t(C + 1) \,. \quad \text{(Eq.10)}$$

En este punto y dependiendo del resto de parámetros del sistema la utilización de un contexto podría dar lugar a la aparición de falsos negativos (notas que están presentes pero que el sistema no detecta), no siendo valido el resultado en piezas musicales cuya ejecución contenga notas de una duración menor a la *l* calculada. Como dato a tener en cuenta, cabe señalar, que la duración de una semicorchea con un tempo de 107 bpm, es de *l* = 139,2ms.

Para finalizar se realiza el proceso de umbralización en el cual los picos de la función de detección que superen un umbral θ serán considerados como *onsets*, siendo el resultado del sistema los instantes de tiempo donde se producen dichos picos.

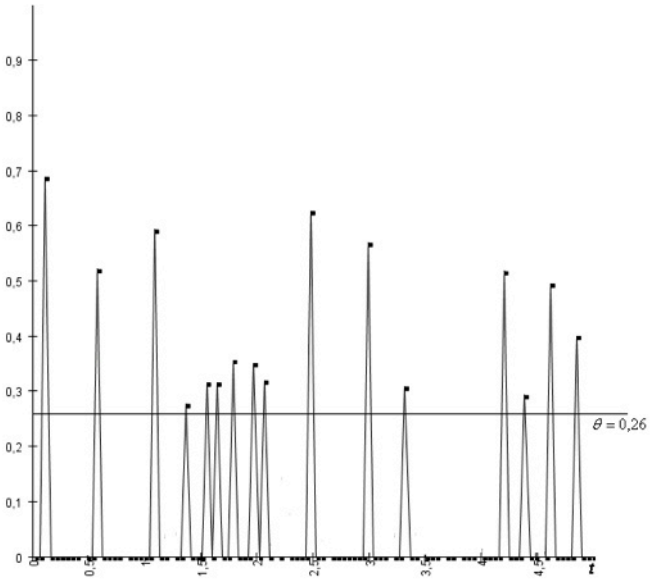
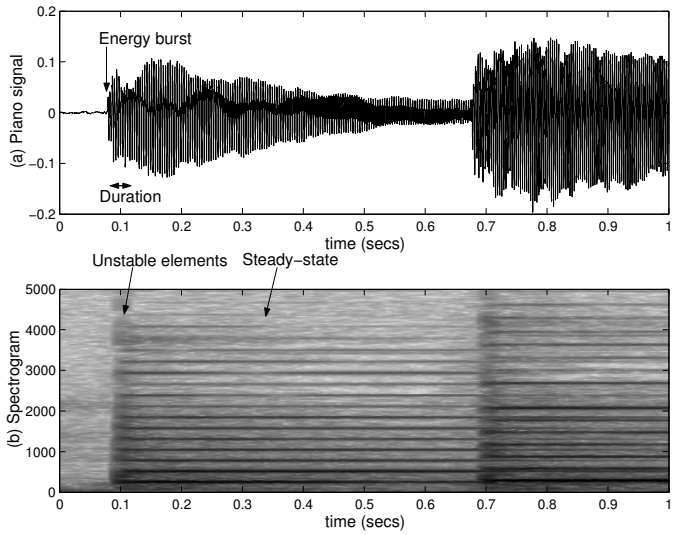


Fig.3.5.- Umbralización de los picos de la función de detección.

4.2 Detección de Onsets en Señales Musicales en el Dominio Complejo.

Este sistema se presenta como un método de detección de onsets mejorado respecto a anteriores aproximaciones, basadas en energía o basadas en fase, combinando ambos tipos de información en el dominio complejo. Esto genera una función de detección en la que se resalta la posición de los onsets y se suaviza todo lo demás. Según los autores los experimentos realizados muestran altas tasas de detección con bajo porcentaje de errores, y lo presentan como un método de detección mas robusto que sus predecesores, tanto en la teoría como en la práctica.



La figura 1 muestra el sencillo caso de dos onsets de piano e ilustra el incremento de energía, la corta duración y la inestabilidad relativa a la parte del transitorio de la nota, así como la estabilidad de la parte estacionaria. La amplia gama de señales y formas de inicio pueden ser mucho más complejas que este ejemplo, pero estos fenómenos son comunes a la mayoría.

Casi todos los algoritmos de detección de onsets se pueden separar en dos partes distintas. La primera de ellas, frecuentemente llamada “función de detección”, convierte la señal a partir sus de muestras en el dominio del tiempo en una función mas efectiva a la hora de localizar los transitorios de onsets. La segunda parte de cualquier algoritmo de detección

de onsets es a menudo llamada “detección de picos”, e implica la localización de puntos en la función de detección que corresponden a los transitorios de inicio de eventos.

Una función de detección robusta presentará normalmente picos muy marcados en los transitorios, y unos cuantos picos espurios localizados fuera de estos. En la mayoría de los casos se usa el algoritmo de detección de picos sobre la función de detección mas robusta posible. Por esta razón los autores centran la mayor parte de este trabajo en la etapa de construcción de la función de detección.

La etapa de detección de picos debe ser eficaz en la selección únicamente de los picos correspondientes a onsets. Por lo tanto, simplemente debería seleccionar todos los picos en el improbable caso de tener una función de detección perfecta. Una umbralización eficaz de la función de detección para ignorar los picos espurios es un problema muy común en la etapa de detección de picos. Aunque se presenta un sistema de detección de picos en este método, se concentra la mayor parte de la investigación en conseguir una función de detección que sea útil casi con cualquier sistema de detección de picos.

Por norma general los esquemas de detección de onsets usan aproximaciones basadas en la energía de las señales y ponderadas por bandas de frecuencia. En los últimos años se han ampliado con estudios que incluyen esquemas de sub-bandas o proponiendo enfoques basados en el estudio de la fase de las señales. Este enfoque ofrece mejoras claras en señales mas suaves, con menos elementos percusivos. Esta idea se desarrollo en anteriores trabajos de estos autores, donde propusieron una aproximación que combinaba el estudio de la energía y la fase. En este trabajo añaden a esta idea el desarrollo de un enfoque en el dominio complejo para la detección de onsets.

4.2.1 Anteriores aproximaciones a la detección de onsets.

A continuación se detallan los métodos existentes anteriores al trabajo que presentan los autores del presente estudio, sobre los que se basan y pretenden mejorar con esta aproximación.

4.2.1.1 Detección de Onsets basada en la Energía.

La aparición de una nueva nota siempre dará lugar a un aumento de la energía de la señal. En el caso de notas percusivas con ataques fuertes como las producidas por tambores, el aumento de esta energía será muy pronunciado. Por esta razón, la energía ha demostrado ser una medida útil, sencilla y eficiente para los transitorios pronunciados y por consiguiente para ciertos tipos de notas. Si tenemos en cuenta el cuadrado de la energía de un instante de la señal x(m):

$$E(m) = \sum_{n=(m-1)h}^{mh} |x(n)|^2$$

donde h es el tamaño de la señal, m el numero de muestras y n la variable de integración. Tomando la primera derivada de E(m) se produce una función de detección que puede ser usada para detectar localizaciones de onsets.Esta es la aproximación mas simple a la detección de onsets. Esta idea se puede ampliar considerando frames de una FFT. Dada una señal en el dominio del tiempo, su STFT viene dada por:

$$S_k(m) = \sum_{n=-\infty}^{\infty} s(n)w(mh-n)e^{-j2\pi nk/N}$$

donde k = 0,1,...,N - 1 es el indice de frecuencia y w(n) es una ventana de longitud finita. De ello se desprende que la diferencia de amplitud es:

$$\delta S = \sum_{k=1}^{N} |S_k(m)| - |S_k(m-1)|$$

4.2.1.2 Detección de Onsets basada en la fase.

De manera intuitiva, el análisis de Fourier propone que una señal puede ser representada por un conjunto de osciladores sinusoidales con variaciones temporales en amplitud, frecuencia y fase. Durante el estado estable de la señal estos osciladores tenderán a tener

amplitudes y frecuencias estables. Por lo tanto, la fase k-ésima del oscilador en un momento dado n podría ser predicha fácilmente de acuerdo con:

$$\tilde{\varphi}_k(n-1) - \tilde{\varphi}_k(n-2) = \tilde{\varphi}_k(n) - \tilde{\varphi}_k(n-1)$$

donde el operador Yk denota una desenvoltura de la fase. Esto implica que la desviación de fase real entre el objetivo y los valores de fase real viene dada por el término:

$$d_\varphi = \text{princarg}[\tilde{\varphi}_k(n) - 2\tilde{\varphi}_k(n-1) + \tilde{\varphi}_k(n-2)]$$

donde los Mapas de Princarg se limitan en el intervalo [-pi, pi]. Dy tenderá a cero si el valor de fase se predijo con exactitud y se desviará en caso contrario. Esto último sucede para la mayoría de los osciladores durante los transitorios de ataque. Esto se puede extrapolar a la distribución de todos los osciladores en un instante de tiempo. Durante el periodo estable de una señal la mayoría de valores se concentrarán en torno a cero creando una distribución abrupta. Por otro lado, durante los transitorios de ataque la distribución será mas amplia y menos pronunciada. De esto se desprende la posibilidad de construir una acertada función de detección mediante la medición de la dispersión de distribución.

4.2.1.3 Aproximación combinada de energía y fase.

Los métodos basados en la energía son sencillos y directos, y por lo tanto ampliamente utilizados, aunque existen casos, especialmente con mezclas de múltiples fuentes cuando la superposición de notas es común, en los cuales la detección se complica. Los enfoques basados en la fase ofrecen una alternativa a esto y aumentan la eficacia para detecciones con inicios menos abruptos. Sin embargo estos métodos son susceptibles a la distorsión de fase y a las variaciones introducidas por la fase de componentes ruidosos.

En un trabajo anterior los autores propusieron un método que combina los enfoques tanto de energía como de fase. Se hizo uso de una conducta similar de la distribución de las

desviaciones de fase y magnitud de las diferencias espectrales. Las medidas de dispersión por frame para cada distribución fueron obtenidas como:

$$\eta(n) = \mathrm{mean}(f_n(|x|))$$

Donde F(x) es la función de densidad de probabilidad del conjunto de datos. Después se multiplicaron, haciendo hincapié en la característica de fase de los componentes más relevantes para el análisis. Este método compensaba las inestabilidad de uno y otro método y producía picos más prominentes para la detección de onsets. Los resultados superaron tanto a las aproximaciones basadas en energía como a las basadas en fase.

4.2.2 Detección de Onsets en el dominio complejo.

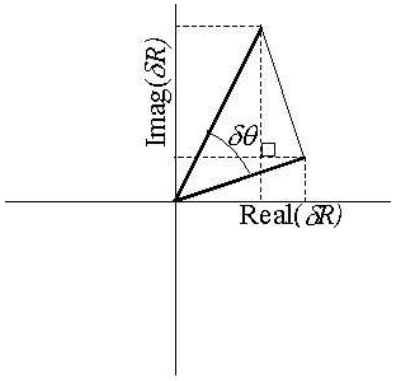


Figura 2: diagrama de fasores en el dominio complejo, la desviación de fase entre el vector objetivo y el común, y la distancia euclidea entre ellos.

Por definición, para regiones localmente estables en señales de audio, se mantiene que la frecuencia y la amplitud permanecen constantes. En secciones anteriores se ha mostrado que mediante la inspección de los cambios en frecuencia y amplitud se pueden localizar transitorios de onsets. Sin embargo, se puede considerar al mismo tiempo el efecto de ambas variables mediante la predicción de los valores en el dominio complejo. Se puede suponer que, en su forma polar, el valor objetivo para una ventana de la FFT está dada por:

$$\hat{S}_k(m) = \hat{R}_k(m)e^{j\hat{\phi}_k(m)}$$

donde el objetivo amplitud Rk (m) corresponde a la magnitud de la trama anterior | Sk (m - 1) |, y la fase φk objetivo (m) se puede calcular como la suma de la fase anterior y la diferencia de fase entre los cuadros anteriores:

$$\hat{\phi}_k(m) = \text{princarg}[2\tilde{\varphi}_k(m-1) - \tilde{\varphi}_k(m-2)]$$

Podemos entonces considerar el valor medido en el dominio complejo de la FFT:

$$S_k(m) = R_k(m)e^{\phi_k(m)}$$

donde Rk y φk son la magnitud y la fase de la ventana de k-ésima de la trama actual de la STFT. Mediante la medición de la distancia euclidiana entre el objetivo y los vectores comunes en el espacio complejo, podemos entonces cuantificar la estacionariedad de la k-ésima trama como:

$$\begin{aligned}\Gamma_k(m) &= \{[\Re(\hat{S}_k(m)) - \Re(S_k(m))]^2 + \cdots \\ &\quad [\Im(\hat{S}_k(m)) - \Im(S_k(m))]^2\}^{\frac{1}{2}}\end{aligned}$$

Sumando esta estacionariedad mediada a traves de todas las K, podemos construir una función de detección frame a frame como:

$$\eta(m) = \sum_{k=1}^{K} \Gamma_k(m)$$

La ecuación 11 puede ser simplificada mediante mapeo de Sk (m) sobre el verdadero eje (forzando φk (m) = 0), de tal manera que:

$$\hat{S}_k(m) = \hat{R}_k(m) = R_k(m-1)$$

Esto implica la rotación de los fasores de la figura. 2, de modo que Sk (m) se puede representar mediante la desviación de fase (Ec. 5):

$$S_k(m) = R_k(m)e^{jd_{\varphi k}(m)}$$

Ahora se considera la diferencia entre esta aproximación de predicción en el dominio complejo y la diferencia mínima de amplitud medida para K-ésima ventana

$$\delta S_k(m) = \hat{R}_k(m) - R_k(m)$$

Con el mapeo sobre el eje real de Sk (m), la ecuación 10 se convierte en:

$$\Gamma_k(m) = \{[\hat{R}_k(m) - \Re(S_k(m))]^2 + \Im(S_k(m))^2\}^{\frac{1}{2}}$$

que se puede desarrollar como:

$$\begin{aligned}\Gamma_k(m) &= \{[\hat{R}_k(m) - R_k(m)cos\big(d_{\varphi k}(m)\big)]^2 + \cdots \\ &\quad [R_k(m)sin\big(d_{\varphi k}(m)\big)]^2\}^{\frac{1}{2}}\end{aligned}$$

expandiendo:

$$\begin{aligned}\Gamma_k(m) &= \{\hat{R}_k^2(m) - 2\hat{R}_k(m)R_k(m)cos(d_{\varphi k}(m)) + \cdots \\ &\quad R_k^2(m)sin^2(d_{\varphi k}(m)) + \cdots \\ &\quad R_k^2(m)cos^2(d_{\varphi k}(m))\}^{\frac{1}{2}} \qquad (17)\end{aligned}$$

y simplificando:

$$\begin{aligned}\Gamma_k(m) &= \{\hat{R}_k(m)^2 + R_k(m)^2 - \cdots \\ &\quad 2\hat{R}_k(m)R_k(m)cos(d_{\varphi k}(m))\}^{\frac{1}{2}}\end{aligned}$$

Para el caso de dφk (m) = 0:

$$\begin{aligned}\Gamma_k &= \{\hat{R}_k^2(m) + R_k^2(m) - 2\hat{R}_k R_k\}^{\frac{1}{2}} \\ &= \hat{R}_k(m) - R_k(m)\end{aligned}$$

Por lo tanto Γk (m) es solamente igual a δSk (m), donde dφk (m) es igual a cero, o cuando la predicción de fase es "buena". En ese caso, se toma en cuenta sólo la diferencia de energía. En el caso de dφk (m) ≠ 0, se está teniendo en cuenta un término adicional de desviación de fase de la predicción.

η (m) constituye una función de detección adecuada mostrando picos abruptos en los puntos de poca estacionariedad. La Figura 3 representa la función de detección de una sección de una señal de guitarra. La figura también proporciona ejemplos individuales de fase y amplitud. El enfoque del dominio complejo es claramente menos ruidoso, por lo tanto, simplifica de la tarea de selección de picos y permite una detección más robusta.

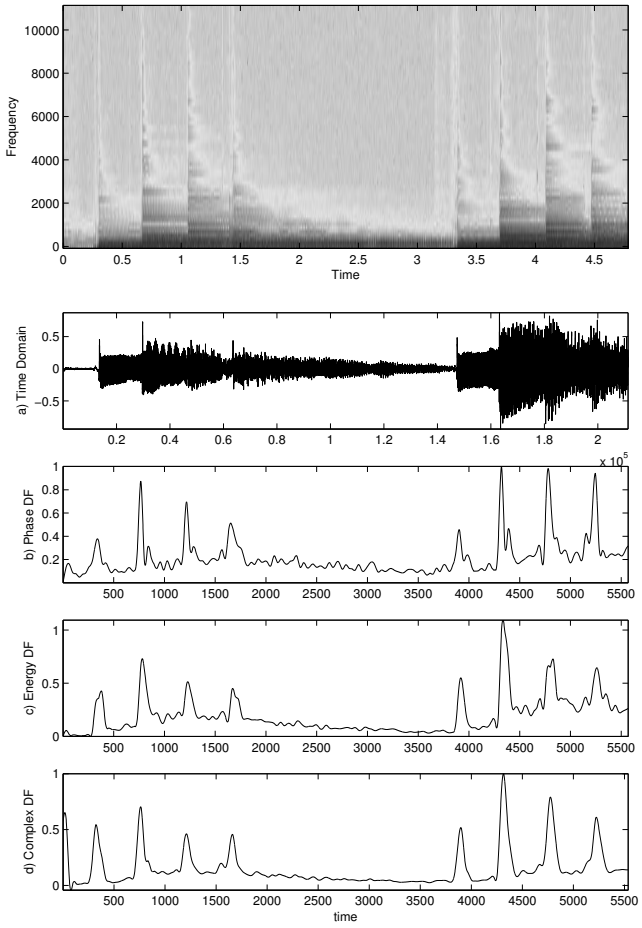


Figura 3: Espectrograma de una señal musical y su correspondiente representación en el dominio del tiempo (a), la función de detección basada en la fase (b), la función de detección de Distribución de Energía (c) y la función de detección de Predicción en el dominio complejo (d).

4.2.3 Detección de Picos.

Umbralizar la función de detección de onsets es un proceso problemático por numerosas razones. En primer lugar, las funciones de detección tienden a ser muy ruidosas, a menos que sean ampliamente filtradas en paso bajo, lo que lleva a un tiempo de resolución mas pobre y a la perdida de los transitorios más débiles. En segundo lugar, las magnitudes de la función de detección tienden a variar considerablemente a lo largo de toda la gama de señales que se presentan en el mundo real. Además, dentro de un segmento corto de la misma señal pueden existir distintos tipos de inicio de notas. Por estas razones, los

umbrales de detección suelen ajustarse manualmente en muchas aplicaciones de detección de onsets. Sin embargo hay muchos casos en los que esto no es práctico. Por ejemplo, en aplicaciones de efectos de audio que requieren la detección de onsets, el usuario no debe ser obligado a seleccionar un umbral de detección para cada señal. Este problema se extiende al considerar aplicaciones en tiempo real. En los algoritmos de detección de picos, el umbral puede establecerse de forma global o local. Mientras los umbrales globales son un método computacionalmente eficiente, los grandes cambios en la dinámica de señal y el contenido dentro de la misma sugieren que la umbralización local es esencial para una detección de onsets efectiva. En este trabajo se utiliza para la detección de picos un simple algoritmo de selección, usando una media móvil ponderada, para determinar la ubicación precisa de los inicios de notas de la función de detección. Este sistema esta basado en el algoritmo de umbralización presentado por I. Kauppinen en "Methods for detecting impulsive noise in speech and audio signals," in *Proc. DSP2002,* donde se utiliza para la detección de ruido impulsivo. El principio básico de este sistema es encontrar la mediana promedio de una señal dentro de una ventana de análisis, por encima de la cual todos los picos se seleccionan como onsets. Cada valor del umbral dinámico δt, para un análisis de desplazamiento de longitud H viene dado por:

$$\delta_t(m) = C_t \,\text{median}\,\gamma_2(k_m), k_m \in [m - \frac{H}{2}, m + \frac{H}{2}]$$

donde Ct es un factor de escala. La Figura 4 ilustra el umbral dinámico para la señal mostrada en la figura 3.

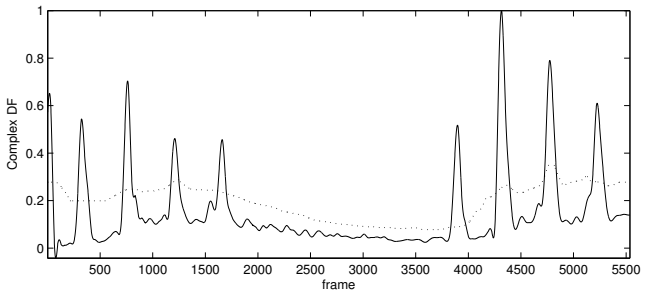


Figura 4: Umbral de mediana adaptativa de una función de predicción de detección en el dominio complejo.

4.2.4 Resultados.

Los experimentos se realizaron sobre una base de datos de una amplia gama de señales musicales polifonicas conteniendo mas de 400 onsets etiquetados a mano. La figura 5 muestra el porcentaje de falsos negativos frente al porcentaje de detecciones correctas para distintos valores de desplazamiento. La curva ideal descansaría sobre el eje y, cuya linea sería una detección 100% correcta. Se puede observar a partir de las curvas generadas, que la curva en el dominio complejo es considerablemente mas robusta en la detección de picos que las curvas que presentan la detección basada solo en fase o en energía.

En la posición óptima de la curva de detección en el dominio complejo, el algoritmo alcanza un porcentaje del 95% de detección correcta frente a un 2% de falsos negativos. Teniendo en cuenta el alcance y la complejidad de las señales musicales utilizados en esta prueba, esto es un resultado muy bueno.

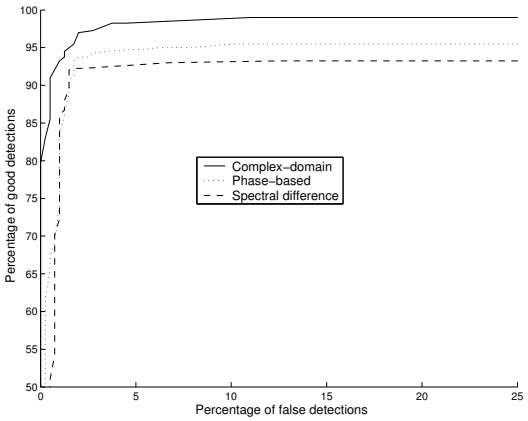


Figura 5: Porcentaje de detecciones correctas frente a porcentaje de falsos negativos para diferentes valores de ponderación y utilizando el método complejo.

4.2.5 Conclusiones.

En general, con los sistemas de detección de onstes basados en la energía se han obtenido buenos resultados para señales de audio con contenido percusivo o inicios de nota "agresivos". Por el contrario, los métodos basados en el estudio de la fase presentan una buena solución para la detección de onsets en señales mas "suaves" como la cuerda frotada. En el dominio complejo, tanto la información de fase como la de energía trabajan juntas, ofreciendo un sistema de detección de onsets, por lo general, mas robusto. Este algoritmo es a la vez fácil de implementar y computacionalmente eficiente. A pesar de esto, es eficaz en una amplia gama de señales de audio.

Puesto que este enfoque en el dominio complejo actualmente tiene un mejor rendimiento en los componentes de baja frecuencia, puede resultar útil dentro de un esquema de multiresolución. Esto presenta la ventaja de que se podrían usar las ráfagas de ruido de alta frecuencia para mejorar la localización temporal de onsets "duros". Puesto que en este caso el análisis debe de ser complejo, sería inadecuado un enfoque basado en wavelets. Sin embargo, análisis de Fourier en multiresolución o wavelets complejos como los descritos en "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters.," in *Proc. IEEE Digital Signal Processing Workshop, DSP 98, Bryce Canyon UT,* 1998 de N.G. Kingsbury, pueden ser útiles para este propósito.

4.3 Normalización Adaptativa para Mejora de la Detección de Onsets de Audio en Tiempo Real.

En este trabajo se describe un nuevo método para el preprocesado de los frames Fase-Vocoder de la STFT para mejorar el desarrollo en tiempo real de un detector de onsets. Este nuevo método se conoce como “Adaptative Whitening”. Este procedimiento consiste en la normalización de la magnitud de cada ventana de acuerdo a un valor máximo reciente para esa ventana, con el objetivo de permitir que cada ventana logre un rango dinámico similar en el tiempo, lo que ayuda a mitigar la influencia de la atenuación progresiva del espectro y la fuerte variación de la dinámica. La normalización adaptativa no requiere de sistemas de aprendizaje, es relativamente ligera computacionalmente, y puede ser ejecutada en tiempo real. Sin embargo, se puede mejorar el rendimiento del detector de onsets en más de un 10 por ciento en algunos casos, y mejorar el desarrollo de la mayoría de detectores de onsets probados.

Los autores de este trabajo presentan resultados que demuestran que la normalización adaptativa mejora significativamente la implementación de varias funciones de detección de onsets basadas en STFT, incluidas funciones basadas en la energía, en flujo espectral, desviación de fase y medidas de desviación complejas. Los resultados arrojan el especial beneficio que muestra el sistema en cierto tipo de señales de audio, por ejemplo mezclas complejas, como en la música pop.

4.3.1 Detectores de onsets actuales.

En los últimos años se han investigado una serie de enfoques a la detección de onsets (J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5): 1035–1047, 2005).

El procedimiento típico implica una etapa de reducción de datos, convertiendo la señal de audio en una función de detección de onsets (ODF) la cual tiene una frecuencia de muestreo mas baja, seguida de una etapa para identificar los onsets en esta ODF.

Los métodos en el dominio del tiempo para producir una función de detección son una posibilidad, pero la mayoría de técnicas actuales convierten la señal al dominio de la frecuencia o al dominio complejo. Típicamente esto se logra usando un vocoder de fase en el que la señal de audio se convierte en un serie de tramas de STFT. Se produce entonces una ODF subsampleada, la cual puede ser por ejemplo una acumulación de un valor único por ventana de la STFT. Los onsets pueden ser entonces seleccionados a través de fenómenos identificables de la ODF, como pueden ser los valores superiores a cierto umbral.

A partir de esta receta básica, se han propuesto muchas variantes. El algoritmo de generación de la ODF merece una consideración importante: en las distintas aproximaciones muchas ODF han sido discutidas, incluyendo algoritmos comunes de potencia espectral, flujo del espectro, contenido en alta frecuencia, desviación de fase, desviación de fase ponderada y desviación compleja.

Véase por ejemplo “J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on on- set detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005. ” y “S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006.” para un estudio sobre estos métodos. En “P. M. Brossier. Automatic Annotation of Musical Au- dio for Interactive Applications. PhD thesis, Queen Mary, University of London, August 2006.” se hace uso de la estadística de la divergencia de Kullback-Leibler y sus variaciones como una ODF. En “A. Lacoste and D. Eck. A supervised classification algorithm for note onset detection. EURASIP Jour- nal on Applied Signal Processing, August 2007.” se entrena una red neuronal para producir una ODF apropiada según información obtenida en el dominio de la frecuencia.

Existen variaciones del esquema básico de detección. Por ejemplo, la señal puede dividirse en bandas de frecuencia separadas, y cada sub-banda tratada por separado, los resultados se combinan posteriormente en una única salida de detección de onsets, como se presenta en “C. Duxbury, M. Sandler, and M. Davies. A hybrid approach to musical note onset

detection. In Pro- ceedings of the DAFx Conference, Hamburg, Ger- many, pages 33–38, 2002."

Hay una variedad de aproximaciones a la detección de onsets que no encaja en la plantilla que se ha descrito. Por ejemplo, en lugar de STFT podemos encontrar aproximaciones con un análisis basado en bancos de filtros, descomposición en wavelets, modelado probabilístico o seguimiento de tono, como en "A. Klapuri. Sound onset detection by applying psy- choacoustic knowledge. In IEEE International Con- ference on Acoustics, Speech and Signal Processing (ICASSP), volume 6, pages 3089–3092, 1999.", "L. Daudet. Transients modeling by pruned wavelet trees. In Proc. International Computer Music Con- ference (ICMC'01), pages 18–21, 2001.", "S. A. Abdallah and M. P. Plumbley. Probability as metadata: Event detection in music using ICA as a conditional density model. In 4th Interna- tional Symposium on Independent Component Anal- ysis and Blind Signal Separation (ICA2003), pages 233–238, April 2003." y "N. Collins. Using a pitch detector for onset detec- tion. In Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 100–106, 2005." respectivamente.

4.3.2 Selección de onsets.

La selección de onsets a partir de la ODF se puede conseguir de diferentes maneras. La mayoría de ODF´s alcanzan un valor alto cuando se produce un onsets. El método básico de selección consiste en el uso de un valor umbral de decisión, valores por encima de ese umbral se consideran onsets. Este método es computacionalmente simple y se puede ejecutar en tiempo real. Este método trabaja bien con ciertas ODF´s, pero puede presentar complicaciones si la ODF es propensa a las variaciones de rango, por ejemplo si la altura de sus picos varía en función de la intensidad general de la música). Como ya propuso Brossier en "P. M. Brossier. Automatic Annotation of Musical Au- dio for Interactive Applications. PhD thesis, Queen Mary, University of London, August 2006", la supresión de componente continua y la normalización pueden mitigar, al menos en parte, este tipo de problemas pero no son adecuadas para usar en tiempo real. Para uso en tiempo real Brossier utiliza un "umbral dinámico" que calcula usando la mediana (opcionalmente la media) a través de un pequeño buffer alrededor del frame actual. Esto puede ser

equivalente a restar una proporción de la mediana de la señal y a continuación utilizar un umbral estático. Una forma alternativa para regularizar la ODF (y compatible con su uso en tiempo real) es aplicar un filtro paso-bajo, pero comparada con el método basado en la mediana puede ser más vulnerable a la influencia indebida de valores extremos como explican en “J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on on- set detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005”.

Numerosos estudios usan un detector de picos para seleccionar los onsets, por ejemplo, “J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on on- set detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005”, “P.M.Brossier,J.P.Bello,andM.D.Plumbley.Real- time temporal segmentation of note objects in mu- sic signals. In Proc. International Computer Music Conference (ICMC’04), pages 458–461, 2004” y “S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006”. Nótese que los algoritmos de detección de picos usados en los detectores de onsets típicamente incluyen algún tipo de sistema de umbralización, para evitar la detección de picos menores que de otro modo podrían causar falsas detecciones. Una de las consideraciones a tener en cuenta para aplicaciones en tiempo real es que el detector de picos requiere al menos un retraso de una ventana de STFT, ya que es necesario determinar que valores alrededor de un pico son mas bajos. En la configuración que los autores usan en este trabajo, por ejemplo, la separación entre ventanas de es de 5,8 ms. Un retardo de 5,8ms puede ser tolerable, sobre todo teniendo en cuenta los limites de resolución del oído humano para discernir acontecimientos temporales, ya que los eventos separados por menos de 30 ms aproximadamente se perciben de manera general como sonidos simultáneos, como se expone en “B. C. J. Moore. An introduction to the psychology of hearing. Academic Press London, UK, 2nd edition, 1982”. Sin embargo, para aplicaciones en tiempo real tales como acompañamiento o resintesis automática, lo ideal sería que todo el sistema respondiera con la simultaneidad percibida. En este contexto, teniendo en cuenta los retardos introducidos por la conversión analógica-digital y digital-analógica y otros procesos que pueden ocurrir en un sentido o en otro, evitar un delay de 5,8ms puede ser oportuno con el fin de permitir que la respuesta

total del sistema sea lo suficientemente rápida para que se perciba todo el proceso simultáneamente a la aparición del onsets.

Por otra parte los tiempos de retardo, aunque imperceptibles pueden afectar a la calidad del sonido, introduciendo en el mismo un efecto tipo "Flam" (técnica de interpretación de batería consistente en tocar una nota muy suave unos pocos ms antes de la nota "real"). La reducción del tiempo de retardo es un objetivo importante en la investigación sobre detectores de onsets.

Una alternativa al uso de un umbral de decisión basado en la selección de picos, es el reconocimiento de patrones. Por ejemplo Kapanci y Pfeffer en "E. Kapanci and A. Pfeffer. A hierarchical approach to onset detection. In Proc. International Com- puter Music Conference (ICMC'04), pages 438–441, 2004" proponen el uso de maquinas de soporte vectorial entrenadas para identificar inicios en una ODF multiresolución. Aunque en teoría los enfoques basados en el aprendizaje pueden ser utilizados para la selección en tiempo real, este estudio no fue dirigido hacia este tipo de aplicaciones.

4.3.3 Elección de un detector de onsets.

La proliferación de los métodos de detección de onsets surge porque no existe ningún método que pueda demostrar ser óptimo de forma general. El rendimiento de un sistema de detección puede diferir dependiendo del tipo de datos de audio utilizados como banco de pruebas: por ejemplo, si la música es en gran parte percusiva o no, y el grado de polifonía. Algunos detectores de onsets están especializados para un dominio en particular (por ejemplo los detectores basados en seguidores de tono, normalmente están destinados a utilizarse con música afinada no percusiva) y algunos (por ejemplo, redes neuronales) pueden especializarse mediante la capacitación con un tipo particular de señal de entrada. Incluso en estos detectores especializados los resultados pueden no ser ideales como debaten en "N. Collins. Using a pitch detector for onset detec- tion. In Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 100–106, 2005".

Además del campo de aplicación, existen prácticas consideraciones que pueden influir en el atractivo de los diferentes métodos de detección de onsets. Los métodos que implican técnicas de aprendizaje por parte de una máquina por lo general requieren una formación con un set de datos en una notación apropiada, lo que puede ser poco práctico. Para aplicaciones en tiempo real, el detector de onsets debe ser causal (no puede depender de momentos posteriores al actual) y debe ser relativamente eficiente, por lo que las decisiones sobre un inicio de evento deben ser producidas con la suficiente rapidez para ser usadas en procesos posteriores, y para que la salida sea percibida como simultánea por un oyente humano, como ya se indico anteriormente.

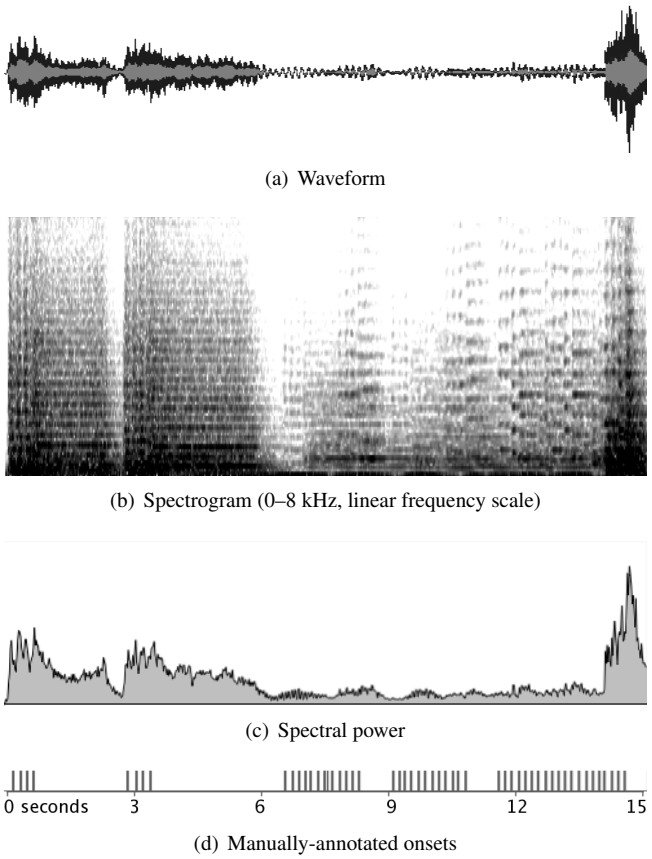
Estas premisas excluyen algunos enfoques, y tienden a favorecer a otros, como las aproximaciones basadas en vocoders, ya que estas se pueden implementar mediante el eficiente algoritmo de FFT, véase "P. M. Brossier. Automatic Annotation of Musical Audio for Interactive Applications. PhD thesis, Queen Mary, University of London, August 2006". La investigación de este grupo de autores está dirigida principalmente a aplicaciones en tiempo real, por lo que están interesados en los desarrollos que puedan mejorar los detectores de onsets basados en STFT sin dejar de ser causales y relativamente eficientes.

4.3.4 Problemas con los detectores de onsets.

No podemos esperar que las detecciones de onsets alcancen siempre el 100% de precisión en relación con las anotaciones "ground truth" proporcionadas por un observador humano, por que una anotación manual realizada por diferentes observadores suelen presentar algunas variaciones, como exponen en "P. Leveau, L. Daudet, and G. Richard. Methodology and tools for the evaluation of automatic onset de- tection algorithms in music. In Proceedings of 5th International Symposium on Music Information Re- trieval, pages 72–75, Barcelona, Spain, 2004". Sin embargo, aún queda mucho por mejorar, sobre todo en los campos que hasta el momento se han probado "difíciles" para los detectores de onsets, como música con un gran rango dinámico y mezclas polifónicas/politímbricas. Los diferentes resultados de los detectores frente a distintas señales de audio en el concurso de detección de audio MIREX son prueba de ello, "Mirex 2006 Audio Onset Detection

Results. www.music-ir.org/mirex2006/index. php/Audio Onset Detection Results, retrieved 30th March 2007".

Como ejemplo de una fuerte variación de la dinámica, se seleccionan los 15 primeros segundos de la Quinta Sinfonía de Beethoven.



(a) Waveform

(b) Spectrogram (0–8 kHz, linear frequency scale)

(c) Spectral power

(d) Manually-annotated onsets

Figure 1. The first fifteen seconds of Beethoven's Fifth Symphony

Se trata de una secuencia de notas muy fuertes seguida de una serie de notas muy suaves. La medida de potencia espectral muestra picos de tamaño considerablemente variable. Esta situación plantea un desafío no solo para los detectores de onsets basados puramente en energía, sino también para otros en los que se estudia un factor de magnitud, incluyendo flujo espectral y desviaciones complejas. Incluso es relevante para ODF´s basadas en fase: para alcanzar buenos resultados la ODF basada en desviación de fase a menudo incluye información de magnitud, ya sea explícitamente como en "J. P. Bello, L. Daudet, S.

Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on on- set detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005" o implícitamente a través de la umbralización de la magnitud de cada frame, como en "S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006".

Normalizar cada ventana de la STFT para fijar una magnitud total puede beneficiar a algunas ODFs, pero claramente no a las ODF de energía, donde el resultado de salida sería un valor fijo. Klapuri en "A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In IEEE International Conerence on Acoustics, Speech and Signal Processing (ICASSP), volume 6, pages 3089–3092, 1999" aplica una ruta alternativa a partir de los principios de la psicoacústica, utilizando la primera derivada del logaritmo de la amplitud. (equivalente al logaritmo de la relación de amplitud entre tramas sucesivas), que produce una ODF independiente de la escala global de amplitud.

Además de la variabilidad temporal, también hay variabilidad a través de las bandas de frecuencia.

Las señales musicales muestran por lo general una atenuación espectral, con las magnitudes máximas que alcanzan los valores más bajos y más baja hacia las bandas de frecuencia más altas. Esto significa que las bandas de frecuencia más bajas pueden "ahogar" las bandas más altas de la ODF tales como la potencia o del flujo del espectro, lo que contribuye mucho más fuertemente a la variación en la señal de ODF; información que pueda estar presente en las bandas más altas se puede despreciar como resultado. La medida HFC se ha mostrado útil en la detección de onsets ("P.M.Brossier,J.P.Bello,andM.D.Plumbley.Real- time temporal segmentation of note objects in mu- sic signals. In Proc. International Computer Music Conference (ICMC'04), pages 458–461, 2004"), y una de las razones es que aproximadamente compensa la atenuación, haciendo hincapié en las contribuciones de las bandas de frecuencia mas altas. Sin embargo la nueva ponderación realizada por HFC no se deriva empíricamente a partir de la pendiente espectral típica de las señales musicales, pero simplemente es lineal en frecuencia. Un forma alternativa para compensar esto, podría ser la de aplicar un filtro de pre-énfasis a la señal de audio.

Tanto HFC como el filtro de pre-énfasis son relativamente “ingenuos” en la manera en que reponderan el espectro de forma fija, independientemente de las características de la señal concreta en estudio. Los autores de este estudio están interesados en encontrar un procedimiento que pueda reponderar el espectro como HFC y pre-énfasis, pero de una forma que sea dependiente de los datos. Proponen así, que esto puede conducir a una mayor precisión de estas técnicas, e idealmente también de una manera que puede mitigar los problemas de las fuertes variaciones dinámicas que pueden confundir a algunos detectores de onsets.

4.3.5 Normalización Adaptativa.

En un primer experimento se analiza un archivo entero de música con un vocoder de fase y se mide la magnitud mas alta que se produce en cada contenedor de frecuencia a través de todo el archivo. A esto se le ha llamado “Perfil Espectral de Pico” (PSP). El PSP podría ser utilizado para nivelar la señal antes de la detección de onsets: dividiendo la magnitud de cada frame de la STFT por el máximo histórico en ese frame, nos aseguramos que cada banda de frecuencia llega a la misma máxima durante la duración de la grabación.

Este proceso hizo mejorar la detección en algunos casos, pero funciono peor en grabaciones con un pronunciado rango dinámico en el tiempo (variaciones de timbre o de dinámica). Esto se puede ver en el ejemplo de Beethoven citado anteriormente. Los picos del espectro en la parte alta serían los que se usan para cambiar la escala, pero no sería apropiado para evaluar la parte mas “tranquila”. Se trata también de un proceso no causal, lo que lo hace inapropiado para los escenarios de detección en tiempo real.

Tres modificaciones introducidas en este procedimiento dan lugar al algoritmo que se presenta. En primer lugar, para la operación de causalidad, solo se consideran las magnitudes de pico a través del archivo de audio hasta el momento actual: el PSP no se deriva en un paso separado sino en el mismo momento que el procesamiento del audio para la detección de onsets. Esto significa que la nueva ponderación espectral solo puede depender de los valores de pico en el pasado con respecto al instante en consideración. En

segundo lugar, con el fin de afrontar mejor el audio en el que la dinámica evoluciona con el tiempo, los valores de PSP decaen exponencialmente con el tiempo, lo que significa que los valores máximos pasados son poco a poco “olvidados”. En tercer lugar, para evitar que los valores de PSP caigan tan bajo que el ruido (como el ruido de cuantización) se sobreamplifique, se añadió un parámetro de “suelo” al algoritmo, por debajo del cual no pueden caer los valores de PSP. El algoritmo iterativo resultante se puede expresar como:

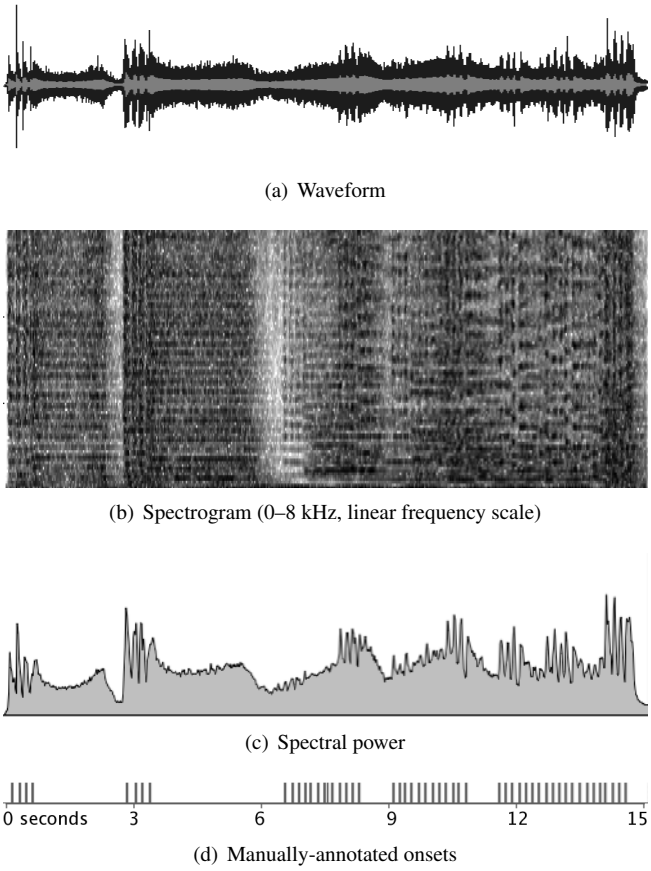
$$P_{n,k} = \begin{cases} \max(|S_{n,k}|, r, mP_{n-1,k}) & \text{if } n > 0, \\ \max(|S_{n,k}|, r) & \text{otherwise.} \end{cases} \quad (1)$$

$$S_{n,k} \leftarrow \frac{S_{n,k}}{P_{n,k}} \quad (2)$$

para n ≥ 0, donde m es el coeficiente de memoria, r el parámetro de suelo, y Sn, k el valor complejo de la STFT en el fotograma de índice n y la frecuencia de índice k.

Por conveniencia, el coeficiente de memoria puede ser calculado de la de la frecuencia de frame de la STFT y del tiempo de relajación de 60dB deseado. es decir, la cantidad de tiempo que le llevaría a un pico caer 60dB. En lo que resta de este estudio en lugar del coeficiente de memoria, se nombrarán los valores correspondientes de este tiempo de relajación. El algoritmo es un proceso de adaptación que tiene como objetivo nivelar la señal en el sentido de llevar la magnitud de cada banda de frecuencia a una gama dinámica similar, de ahí el termino “Adaptative Whitening”. La Figura 2 muestra la misma señal que la Figura 1 pero después del proceso de adaptación. la forma de onda (que aquí se genera mediante el uso de la FFT inversa para convertir la señal adaptada de nuevo al dominio del tiempo) presenta un efecto de normalización similar en apariencia al efecto de un compresor de rango dinámico. El espectrograma muestra el efecto del proceso de adaptación en el dominio de la frecuencia: así como la reducción de la diferencia entre notas fuertes y suaves, la atenuación espectral se elimina a fondo de cada nota, dejando una señal con un perfil espectral generalmente plano. El efecto también se manifiesta en la medida de la potencia espectral, donde notas fuertes y suaves muestran patrones de actividad muy similares. Hay que tener en cuenta que este cambio en la medición de potencia espectral no es necesariamente el mismo que en un proceso de normalización de

amplitud, tal como la compresión de rango dinámico, debido a que se modifica la contribución relativa de las diferentes bandas de frecuencia, así como la amplitud global.



(a) Waveform

(b) Spectrogram (0–8 kHz, linear frequency scale)

(c) Spectral power

(d) Manually-annotated onsets

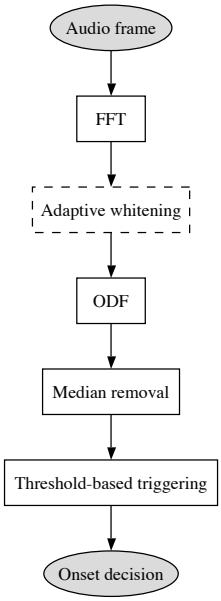
Figure 2. The first fifteen seconds of Beethoven's Fifth Symphony, after adaptive whitening has been applied. Parameters: relaxation time 250 s, floor coefficient 10^{-4}.

Este algoritmo es relativamente ligero de calcular en el hardware actual, ya que solo utiliza operaciones de suma, resta, multiplicación y división en coma flotante. En procesadores típicos estas operaciones solo requieren una instrucción, en comparación con operaciones trigonométricas y logarítmicas, ver "D. Patterson, J. Hennessy, D. Goldberg, and K. Asanovic. Computer Architecture: a quantitative approach. Morgan Kaufmann, 3rd edition, 2003".

Los requisitos de memoria también son pequeños, el requisito principal es la PSP, una matriz de valores en coma flotante del mismo tamaño que el número de ventanas de la STFT.

4.3.5.1 Evaluación.

A fin de probar el efecto “adaptative whitening” en la detección en tiempo real, se investigo en un algoritmo de detección donde el sistema de adaptación pudiera estar o no presente, y se pudieran usar distintos algoritmos basados en energía, HFC, flujo espectral, etc, para generar la ODF. El algoritmo se ilustra en la Figura 3.



Aparte de la elección de la ODF y el uso o no del “adaptative whitening”, el resto de parámetros y partes del sistema se mantienen fijos. La entrada del sistema es una señal de audio monofónico a 44.1Khz, y la FFT se desarrolla usando bloques de 512 muestras con un solapamiento del 50%. Para la selección de onsets, para extraer la mediana de la ODF se consideran los 11 frames previos al actual (este valor se deduce de experimentos previos para determinarlo) y se aplica un umbral de decisión. Los parámetros r y m se establecen también mediante experimentación. Valores típicos para estos parámetros son del rango de 10^{-6} hasta 0.2 para r, y entre 22 y 446 segundos para el tiempo de relajación (la escala de r es relativa a la amplitud global de señal y alcanza un valor máximo de 1). Para realizar los experimentos sin tener que variar parámetros del sistema de adaptación, en los

experimentos se fijo 0.1 para r y 25.6 s para el tiempo de relajación, que generalmente proporcionan buenos resultados aunque no son la configuración óptima para algunos tipos de música.

El set de datos usado para la evaluación fue el banco de canciones etiquetadas manualmente usado en la competición de detección de MIREX 2005 y 2006, dividido en cuatro categorías principales:

- 30 grabaciones monofónicas de instrumentos con tono, incluida la voz.
- 30 grabaciones de percusión, incluidos kits de batería.
- 10 grabaciones polifónicas de instrumentos con tono.
- 15 grabaciones de mezclas complejas incluida música pop, clásica y músicas del mundo.

La longitud de estas muestras oscila entre los 2 y los 36 segundos. Cada grabación esta acompañada de anotaciones realizadas a mano por 3 oyentes diferentes, excepto las mezclas complejas las cuales tienen 5 anotadores diferentes. En total incluyen 9.333 onsets anotados. La evaluación del sistema de onsets es similar a la descrita en otros estudios como “J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on on- set detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005” y “S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006” y que ya ha sido descrita en esta memoria.
Para determinar la utilidad del “Adaptative Whitening” como un paso del procedimiento en este estudio han usado varios procesos de genración de la ODF, en los cuales se ha probado la activación y desactivación del proceso de adaptación. Se han usado las siguientes ODFs:

1. Power (Pow)
2. Phase deviation (PD)
3. Weighted phase deviation (WPD)
4. Rectified spectral flux (SF)
5. Complex deviation (CD)

6. Rectified complex deviation (RCD)
7. High-frequency content (HFC)
8. Modified Kullback-Leibler divergence (MKL)

La generación de estas ODF corresponden a los estudios “S. Dixon. Onset detection revisited. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06),* pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006” para ODfs 2-6, “P. Masri. *Computer Modeling of Sound for Trans- formation and Synthesis of Musical Signals.* PhD thesis, University of Bristol, UK, 1996” para ODf basada en HFC y “P. M. Brossier. *Automatic Annotation of Musical Au- dio for Interactive Applications.* PhD thesis, Queen Mary, University of London, August 2006” para Modified Kullback-Leibler divergence (MKL).

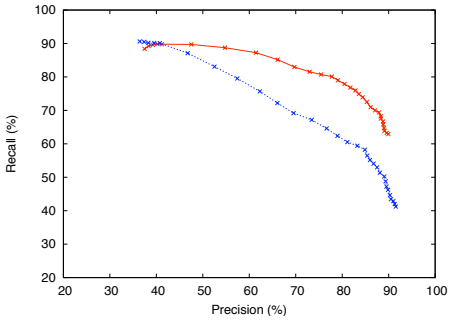
Los resultados obtenidos son los siguientes:

Dataset	ODF	% F-measure	% Precision	% Recall
Complex mixture	Pow	70.2	73.4	67.2
	PD	75.0	77.4	72.8
	WPD	65.5	63.8	67.4
	SF	67.2	81.1	57.3
	CD	72.4	77.2	68.1
	RCD	64.0	61.9	66.2
	HFC	73.5	77.3	70.0
	MKL	78.5	80.9	76.2
Solo drums	Pow	92.8	92.6	93.0
	PD	90.8	91.0	90.7
	WPD	92.2	92.1	92.4
	SF	89.9	94.7	85.5
	CD	93.7	93.9	93.4
	RCD	90.8	88.7	93.0
	HFC	87.7	79.9	97.3
	MKL	94.7	95.0	94.6
Monophonic pitched	Pow	53.7	54.2	53.2
	PD	58.6	66.1	52.6
	WPD	51.7	57.0	47.3
	SF	56.7	57.2	56.2
	CD	57.9	63.1	53.5
	RCD	47.6	50.2	45.2
	HFC	56.8	58.3	55.4
	MKL	64.6	64.9	64.3
Polyphonic pitched	Pow	87.6	90.3	85.1
	PD	81.7	89.6	75.0
	WPD	76.8	72.8	81.3
	SF	75.4	85.2	67.6
	CD	88.6	89.7	87.5
	RCD	68.9	59.8	81.3
	HFC	84.4	85.8	83.0
	MKL	82.4	88.4	77.2

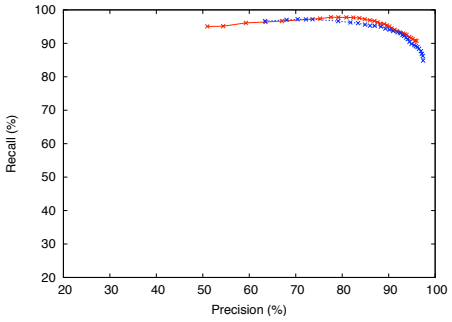
Table 1. Performance at peak F-measure for the onset detector using various ODFs, without adaptive whitening.

Dataset	ODF	% F-measure	% Precision	% Recall
Complex mixture	Pow	79.3	82.8	76.0
	PD	74.9	75.4	74.3
	WPD	80.6	85.3	76.4
	SF	73.7	81.7	67.1
	CD	81.7	82.0	81.4
	RCD	80.3	83.5	77.2
	HFC	76.1	82.5	70.6
	MKL	75.8	77.6	74.0
Solo drums	Pow	93.3	96.0	90.7
	PD	91.8	94.9	88.9
	WPD	93.3	96.5	90.2
	SF	92.5	94.0	91.1
	CD	93.5	96.0	91.1
	RCD	93.5	95.1	92.0
	HFC	90.3	88.0	92.6
	MKL	90.8	90.9	90.7
Monophonic pitched	Pow	66.3	70.8	62.5
	PD	60.6	63.9	57.7
	WPD	60.8	63.0	58.7
	SF	63.3	64.0	62.6
	CD	67.3	73.1	62.4
	RCD	62.0	63.4	60.6
	HFC	62.7	64.9	60.6
	MKL	55.4	51.7	59.6
Polyphonic pitched	Pow	88.0	90.7	85.4
	PD	79.1	80.6	77.6
	WPD	83.8	84.3	83.4
	SF	79.4	85.8	73.9
	CD	87.3	88.1	86.4
	RCD	84.6	83.8	85.5
	HFC	77.4	73.9	81.2
	MKL	62.1	57.0	68.2

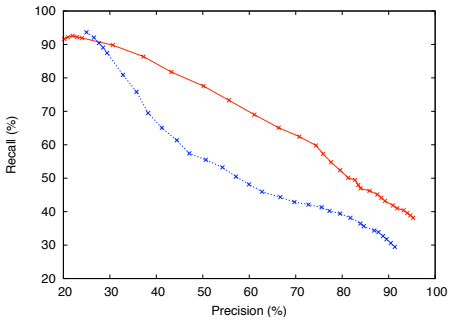
Table 2. Performance at peak F-measure for the onset detector using various ODFs, with adaptive whitening activated.



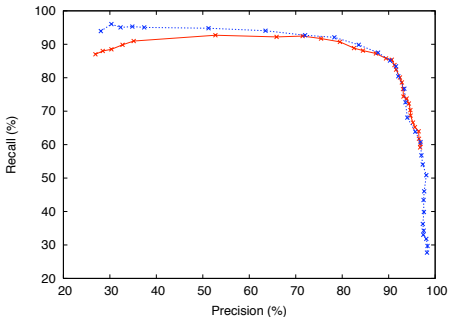
(a) Complex mixture



(b) Solo drums



(c) Monophonic pitched



(d) Polyphonic pitched

Figure 5. Precision/recall plots comparing power-based onset detection with and without adaptive whitening, for each of the four MIREX datasets. The x-axis shows the precision, and the y-axis the recall; the closer to the "top right", the better the results. For each audio type, the "plain" power ODF results at various threshold settings are displayed as the blue (dashed) line, and the equivalent results for the power ODF with adaptive whitening are displayed as the red (solid) line.

4.3.5.2 Discusión.

Los resultados muestran un patrón general de mejora en los resultados de la mayoría de ODfs con el uso del sistema de adaptación, especialmente en mezclas complejas y conjuntos de datos monofónicos afinados. El sistema beneficia mas de manera mas intensa a las ODFs basadas en enegía, WPD, SF, CD y RCD. El rendimiento de HFC se mejoró ligeramente con la excepción de datos polifónicos con afinación, mientras que el rendimiento del sistema bajo PD mejora o empeora ligeramente dependiendo de los datos a examen.

De las ODFs no adaptadas, MKL es la que produce algunos de los mejores resultados e incluso empeora con el uso de adptación. Esta es una excepción interesante que podría se investigada en futuros trabajos. Las funciones RCD y WPD fueron presentadas como una mejora de las funciones CD y PD respectivamente en "S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006". En los resultados no se aprecia esta mejora, e incluso WPD muestra peores resultados que PD con la adaptación activada.

De manera general, según los autores de estos experimentos su mejor configuración del detector de onsets sería CD con "Adaptative Whitening", ODF basada en energía con "Adaptative Whitening", RCD con "Adaptative Whitening" y MKL sin "Adaptative Whitening". Dada la simplicidad del algoritmo los mejores resultados los produjo el sistema de ODF de energía con "Adaptative Whitening". Los autores subrayan los estudios de Dixon respecto a la elección de una ODF basada en factores como la simplicidad de implementación y la velocidad de ejecución. Para el uso en tiempo real la eficiencia en la ejecución es un factor importante, por lo que en este caso la mejor parada sería la ODF de energía con "Adaptative Whitening", como muestra el uso de CPU de la figura 4.

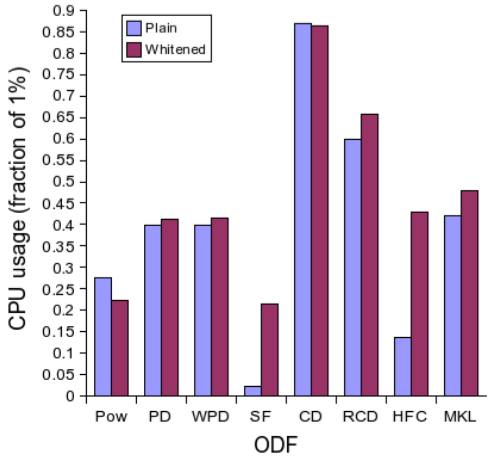


Figure 4. Real-time CPU usage for each ODF, with and without adaptive whitening. The values are derived for each ODF separately, as follows: 60 of the same onset detector are run in parallel, analysing the same 0.25 s audio loop (played with a different phase offset and separate FFT for each ODF instance), and the average CPU usage is recorded. The average CPU usage for the same system without any ODFs is subtracted (i.e. for running the audio playback and FFT), and the resulting value is divided by 60 to give an estimate for the CPU usage of a single ODF. Tests were performed in SuperCollider 3, using the first author's C++ implementation of the ODFs, on a 2 GHz Mac Intel Core 2 Duo.

Algunos investigadores encuentran beneficioso el uso de diferentes ODFs combinadas, creando un sistema híbrido de detección de onsets que puede superar a sus componentes constructivos por separado. Esto podría llevar a investigaciones como el uso en mismo detector de la combinación de una ODF con "Adaptative Whitening" con otra sin adaptar, por ejemplo MKL no adaptada. Otra vía de investigación futura sería explorar completamente el efecto de los parámetros del sistema de adaptación según el tipo de señal musical a estudio.

4.3.5.3 Conclusiones.

El sistema de adaptación presentado por estos autores es un proceso simple y computacionalmente eficiente que mejora el rendimiento de ciertos tipos de detectores de onsets en tiempo real. Modifica de forma adaptativa las magnitudes de las ventanas de la STFT de una manera que compensa la variación de la atenuación espectral y la variación

dinámica de una señal musical. Este sistema mejora de forma particular el rendimiento en detección de mezcla compleja y datos monofónicos afinados, entre 2 y 16 puntos porcentuales, y no mejora los basados en ODFs de desviación de fase ni la modificación de Kullback-Leibler. Se puede decir en este sentido, que el sistema "Adaptative Whitening" hace mas fáciles los casos difíciles. Dadas las similitudes generales en el rendimiento máximo, para su uso en tiempo real según su eficiencia computacional, los autores concluyen que el sistema idóneo sería el uso de una ODF basada en energía con "Adaptative Whitening".

4.4 Separación de fuentes de Kits de Batería usando una Función de Detección de Percusión y Modulación Espectral.

Esta técnica en el dominio de la frecuencia incluye la identificación de la presencia de una señal de tambores usando una nueva función de detección (ODF) de características percusivas, después de la cual se estima la magnitud del espectro en "tiempo-corto" y se escala de acuerdo a la estimación de una función tiempo-amplitud derivada de una medida percusiva.

Cuando la música se compone de sólo tambores, algunos algoritmos existentes proporcionan resultados razonablemente precisos, sin embargo, con la presencia de instrumentos afinados, los algoritmos se vuelven menos robustos y menos exactos a través de una falsa detección del ritmo y golpes que faltan por completo. Un algoritmo de separación de tambores en este caso sería un pre-proceso viable con el fin de superar algunos de los problemas asociados con la transcripción de tambores en presencia de instrumentos con tono.

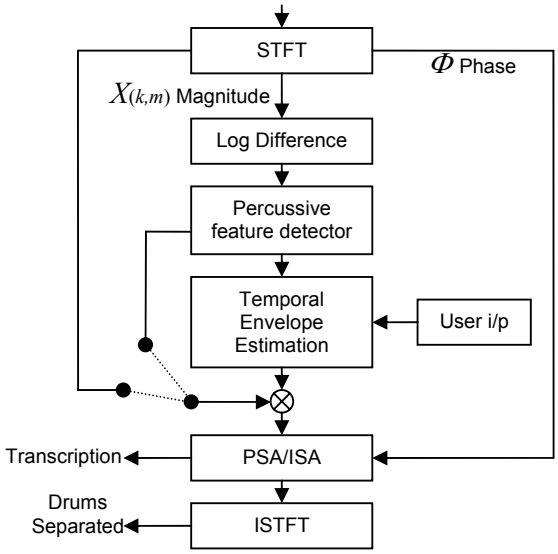
En este trabajo Dan Barry, Derry Fitzgerald, Eugene Coyle y Bob Lawlor, presentan un modo rápido y eficaz para la descomposición de un espectrograma utilizando una técnica simple que implica la detección de caracteristicas percusivas y modulación espectral, que se traduce en la extracción de las partes de batería de una mezcla polifónica.

4.4.1 Metodología.

La mayoría de los kits de batería usados en la música popular se caracterizan por un rápido aumento de la energía de banda ancha seguida por un rápido decaimiento. Esto es particularmente cierto para los sonido de bombo y caja que se pueden considerar los mas comunes en la música moderna. Por otro lado, los instrumentos afinados solo muestran energía en múltiplos enteros de la notas fundamentales que se tocan en este tipo de música.

Con esto en mente los autores desarrollan un detector de onsets que no se ocupa de la medición de aumentos rápidos de energía, sino más bien de un detector que mide la

naturaleza del ancho de banda o la "percusividad" de la aparición, independientemente de la energía real presente. De esta manera se pueden detectar golpes de tambor de diferentes velocidades. Del análisis de cada trama de una STFT se deriva un perfil temporal de percusión y se asigna una medida de percusión en función del mismo. Cada frame se escala de acuerdo a esta medida. Debe considerarse entonces que las regiones del espectrograma con medidas de percusión bajas se reducirán significativamente. En la resíntesis solo quedarán las regiones de percusión. El espectograma es modulado por una envolvente que corresponde a la percusión detectada en la señal.



La figura ilustra el funcionamiento general del algoritmo. Se toma la magnitud de la STFT de la señal y la información de fase se mantiene para propósitos de resintesis mas adelante. A continuación se calcula la diferencia logarítmica de cada componente de frecuencia entre las tramas consecutivas. Esta medida indica la rapidez de fluctuación del espectrograma. Si la diferencia del registro supera un umbral definido por el usuario, se considera que nos encontramos ante un inicio de nota de percusión y se incrementa un contador. El valor final de este contador, una vez se ha analizado cada valor de banda de frecuencia, será la medida de percusividad de la trama actual. Una vez procesados todos los frames, tenemos un perfil temporal que describe las características de percusión de la señal. Este perfil se usa entonces para modular el espectrograma antes de la resíntesis.

4.4.2 Estimación temporal.

En primer lugar se toma una STFT de la señal según:

$$X(k,m) = abs\left[\sum_{n=0}^{N-1} w(n)x(n+mH)e^{-j2\pi mk/N}\right] \quad (1)$$

donde X (k, m) es el valor absoluto de la STFT compleja dada en la ecuación 1 y donde m es el índice de frame temporal, k es el índice de banda de frecuencia, H es el tamaño de salto entre ventanas y N es el tamaño de la ventana FFT, donde w (n) es una ventana ideal de longitud N también. A continuación se toma la diferencia logarítmica del espectrograma con respecto al tiempo como en la ecuación 2.

$$X'(k,m) = 20\log_{10}\frac{X(k,m-1)}{X(k,m)} \quad (2)$$

para todo m y $1 \leq k \leq K$.

Con el fin de detectar la presencia de una batería se define una medida percusiva dada por la ecuación 3.

$$Pe(m) = \sum_{k=1}^{K}\begin{cases} P(k,m)=1 & if\ X'(k,m) > T \\ P(k,m)=0 & otherwise \end{cases} \quad (3)$$

Donde, T es un umbral que significa el aumento de la energía medida en dB que debe ser detectado dentro de un canal de frecuencia antes de que se considera que es un onset de percusión. Efectivamente la ecuación 3 se comporta como un contador, Pe (m) es simplemente un recuento de la cantidad de bandas que son positivas y superan el umbral. P (k, m) contiene un 1 si la condición de umbral se cumple y por lo demás un cero. Hay

que tener en cuenta que la energía real presente en la señal no es importante en este caso, simplemente queremos una medida de la "banda ancha" o cuan percusivo es el onset. La siguiente figura muestra la eficacia de este enfoque. Detectores de inicio basados en energía estándar, tales como [8] no será capaz de distinguir entre los inicios de banda estrecha y de banda ancha. En estos sistemas, el nivel de detección será intrínsecamente ligado a la energía de la señal en un momento dado. La función de detección que hemos descrito es independiente de la energía y por lo tanto puede hacer frente a onsets de baja energía, siempre y cuando sean de banda ancha.

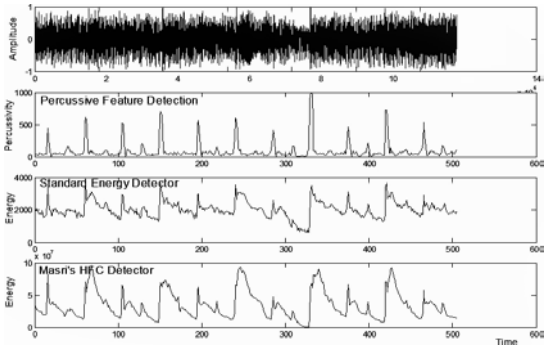


Figure2: The top plot shows the original audio clip. Plot 2 shows our percussive onset detector. The third plot shows the standard energy detector and the bottom plot shows Masri's high frequency weighted detection function [8]

Notar que la función de detección de característica percusiva que ha sido descrita incluso puede detectar los golpes de hi-hat de baja amplitud entre los eventos de bombo y caja.

4.4.3 Modulación espectral.

Ponderando la percusividad Pe(m) de cada frame, el espectrograma es modulado en consonancia con la percusión. Esto resulta en que la salida del algoritmo sólo se activa en la presencia de un sonido de batería. Hay algunas opciones cuando se trata de re-síntesis, lo más sencillo es simplemente multiplicar el frame original por la medida de percusión:

$$Y(k,m) = Pe(m)^{\Psi} X(k,m) \qquad (4)$$

para todo m y $1 \leq k \leq K$.

Con el fin de controlar las características de decaimiento de la envolvente de percusión simplemente elevamos la medida de percusión, Pe (m), a la potencia de Ψ. Los valores más altos de Ψ conducirán a un decaimiento más rápido. El parámetro se establece por el usuario en base a que se logren resultados satisfactorios en audición. La ecuación 4 resulta en una separación temporal de las señales de batería pero no una separación de frecuencias. Otras fuentes que estaban presentes en el mismo instante de tiempo que la batería también estarán presentes, pero decaen cuando decae la batería. Este método es particularmente útil para variar el nivel de la batería dentro de una señal mezclada. Para ello, la señal de batería separada se añade a la señal original en la proporción deseada. Este proceso permite un control mucho mayor sobre el rango dinámico de una señal, que las técnicas de compresión dinámica estándar.

La otra opción para la resíntesis que desacopla los tambores de la mezcla, tanto en el tiempo y dominio de la frecuencia es la siguiente:

$$Y(k,m) = Pe(m)^{\Psi} X(k,m) P(k,m) \qquad (5)$$

Al multiplicar el frame de la máscara binaria P (k, m), sólo estamos resintetizando componentes de frecuencia que estaban presentes durante el inicio de percusión. Esto altera el timbre un poco, pero se suprimen de manera efectiva fuentes no percusivas en la mezcla.

La señal de bateria separada se resintetiza a continuación, utilizando el espectro de magnitud modulado con la información de la fase inicial, ecuación 6. Se ha demostrado en "Barry, D., Lawlor, R. and Coyle E., "Comparison of Signal Reconstruction Methods for the Azimuth Discrimination and Resynthesis Algorithm", Proc. 118th Audio Engineering Society Convention, May 28-31, Barcelona, Spain, 2005" que el uso de la información de fase de la mezcla original es más preciso que el uso de una aproximación de error al cuadrado tal como muestran en "Griffin D. W., Lim J.S., "Signal Estimation from Modified Short-Time Fourier Transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-32, no. 2, April 1984".

$$y(n+mH) = w(n)\left(\frac{1}{K}\sum_{k=1}^{K} Y(k,m)\cdot e^{j\angle x_{\omega}(k,m)}\right)^{norm} \quad (6)$$

La salida debe ser normalizada debido al hecho de que la magnitud de los frames se ha reducido de acuerdo con la medida de percusión. w (n) es una función de ventanas de síntesis que se requiere para mantener una transición suave en los límites de la trama, ya que el proceso va a alterar la magnitud del espectro de tiempo corto. Puesto que es tanto una ventana de análisis y síntesis, es necesario el uso de un 75% de solapamiento con el fin de tener una reconstrucción de suma constante.

4.4.4 Resultados.

El algoritmo ha sido aplicado a muchas grabaciones populares y logra separaciones de alta calidad en la mayoría de los casos. La siguiente figura muestra la separación que se ha conseguido en una pieza típica de la música rock.

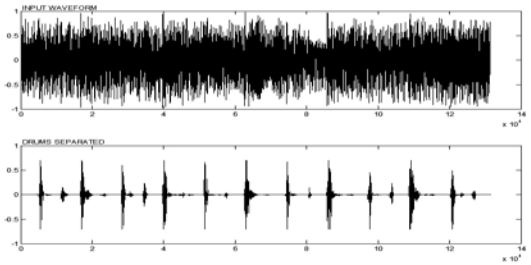


Figure 3: The plot shows the original input file and the drum separation which resulted.

Los tambores son distinguibles mediante inspección visual en el dominio de tiempo en la figura. La función de detección de percusión ha discriminado con precisión entre los eventos de batería y el resto de eventos. La salida del detector se usa entonces para modular el espectrograma que se invierte para producir el diagrama inferior, que es una reconstrucción en el dominio del tiempo de los eventos de batería presentes en la señal.

Para demostrar la utilidad del algoritmo como una etapa de tratamiento previo antes de realizar la transcripción de batería, una prueba informal se llevó a cabo en una pieza de alta compresión de audio que es un "peor escenario" para los algoritmos de transcripción batería.

La compresión de la que hablamos es la compresión de rango dinámico como opuesta a la compresión de reducción de la tasa de bits. Este tipo de compresión se utiliza para aumentar el nivel medio del audio y se aplica a muchas grabaciones modernas en una etapa conocida como masterización. Se reduce eficazmente los niveles de pico y se aumenta el nivel RMS dinámicamente, por lo que es particularmente difícil para la transcripción basada en técnicas tales como las mostradas en "FitzGerald, D., Coyle E, Lawlor B., "Sub-band Independent Subspace Analysis for Drum Transcription", Proceedings of the. Digital Audio Effects Conference (DAFX02), Hamburg, pp. 65-69, 2002" y "FitzGerald, D., Coyle E, Lawlor B., "Drum Transcription in the presence of pitched instruments using Prior Subspace Analysis" Proc. Irish Signals and Systems Conference 2003, Limerick. July 1-2 2003" para distinguir las baterías. El algoritmo de separación se aplicó a esta grabación. Análisis Previo de Subespacio (PSA) mostrado en [3], es una técnica para la transcripción de baterías que se aplicó a ambos espectrogramas no procesados y separados. Los resultados obtenidos se muestran en las Tablas 1 y 2. Se puede observar que el uso del algoritmo de separación ha aumentado sustancialmente el rendimiento del algoritmo de PSA en la transcripción de baterías en la presencia de instrumentos afinados. Los porcentajes se obtienen utilizando las siguientes medidas:

$$correct = \frac{total - undetected - incorrect}{total} \cdot 100$$

Type	Total	Missing	Incorrect	%
Snare	5	2	7	-80
Kick	6	1	2	50
Overall	11	3	9	-9

Table 1: Drum Transcription obtained using PSA on the unprocessed signal

Type	Total	Missing	Incorrect	%
Snare	5	0	0	100
Kick	6	0	1	83
Overall	11	0	1	91

Table 2: Drum Transcription obtained using PSA after the drum separation algorithm

En la tabla 1, el porcentaje de detección general es -9%. Esto es debido al hecho de que el algoritmo de PSA hizo varios falsos positivos, es decir, detecta eventos que no corresponden a eventos de batería. 2 de 5 golpes de caja se perdieron y 1 de 6 golpes de bombo se perdieron junto con varios falsos positivos para ambos. Los resultados de la Tabla 2 muestran claramente que el algoritmo de PSA se ha beneficiado enormemente de la técnica de separación descrita en este artículo. No hay eventos perdidos, sólo había un falso positivo en el caso del bombo.

La técnica Análisis Subespacial Independiente (ISA) [2] también se beneficia enormemente cuando el algoritmo de separación que se presenta aquí se utiliza como un pre-proceso. El diagrama siguiente muestra las diferencias entre la aplicación de ISA directamente al audio sin procesar, Figura 4, y la aplicación de ISA para el espectrograma separado, Figura 5.

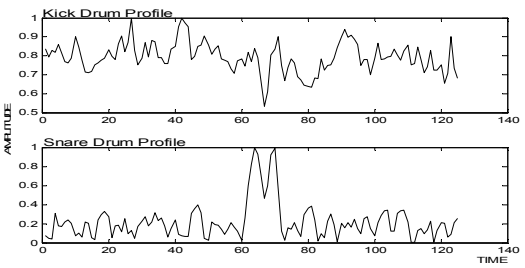


Figure 4: ISA was applied directly to the same audio clip shown in figure 3.

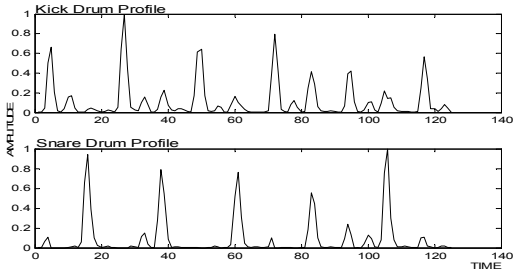


Figure 5: ISA after the separation algorithm has been applied

4.4.5 Conclusiones.

Los autores de este trabajo han presentado un sistema capaz de separar las fuentes de batería de una mezcla polifónica. El algoritmo es útil en el contexto de procesamiento de audio para la producción de música y la educación. También han ilustrado que el uso de este algoritmo como una etapa de pre-procesamiento para los algoritmos de transcripción de batería mejora en gran medida los resultados de transcripción.

4.5 Segmentación Temporal para Anotación de Música y Audio en Aplicaciones Interactivas.

Basado en el trabajo de otros investigadores, Paul Brossier ha reunido en una biblioteca dedicada, una colección de técnicas y algoritmos que permiten estudiar, analizar y manipular señales de audio digital. Aubio, es una herramienta diseñada para la extracción de anotaciones de señales de audio, sus características incluyen la detección de tono, seguimiento y análisis de ritmo, creación de MIDI a partir de audio y por supuesto la función que investigamos en este proyecto y que es la piedra angular de todas estas funciones: la segmentación del audio en eventos de inicio de notas, onsets. Todas estas características han sido implementadas de manera causal para permitir su uso en tiempo real, por ejemplo de manera on-line en efectos de audio o instrumentos virtuales, off-line en editores de audio y softsamplers.

En este proyecto nos centraremos en el estudio del detector de onsets de Aubio en su versión plug-in para Vamp de Sonic Visualizer, que se encuentra explicado en el Capitulo 2 de la tesis de Paul M. Brossier bajo la dirección de Dr. Mark Plumbley, Prof. Mark Sandler, Prof. Eduardo R. Miranda y Dr. Michael Casey en el Centro para Música Digital Queen Mary de la Universidad de Londres: “Automatic Annotation of Musical Audio for Interactive Applications”.

En la segunda parte de dicho capitulo, se describen un número de aproximaciones a la detección de onsets en audio musical, desde técnicas temporales a bancos de filtros y métodos estadísticos, algunas de ellas ya mencionadas en esta Trabajo Fin de Grado. Como norma general, se ha visto que estas aproximaciones pueden separarse en dos cuestiones principales: la construcción de una ODF para caracterizar los cambios en la señal, y la detección de picos en esa función para extraer los tiempos de onsets perceptiblemente relevantes. Para cumplir los requisitos necesarios en aplicaciones en tiempo real, se aborda específicamente una solución de baja latencia para el proceso de selección de picos. El objetivo del detector de picos que se implementa en este estudio es reducir al mínimo el retardo y lograr precisión temporal, dos restricciones requeridas para aproximarse a la capacidad de respuesta del oído humano.

4.5.1 Modelos perceptuales para segmentación temporal. Funciones de detección de onsets Fase-Vocoder.

En este punto estudiaremos las ODFs que implementa Aubio en su detector de Onsets. Para ello se utiliza un vocoder de fase para obtener una representación tiempo-frecuencia de la señal. El vocoder de fase y su uso para las señales musicales han sido descritos en detalle en la literatura [Portnoff, 1976, Moorer, 1978, Dolson, 2001, de Ir Tzen et al., 2000]. Las notaciones que utilizan a lo largo de este documento son los siguientes: para una señal x en el tiempo n, definimos X [n] como su Short Time Fourier Transform (STFT). Xk [n], el valor de la componente espectral compleja en el k-ésimo contenedor en n, se puede expresar en su forma polar como | X k [n] | ejφk [n], donde | X k [n] | es la magnitud espectral del contenedor, y φk [n] su fase. El tamaño típico de ventana para cada vocoder de fase de 1024 o 512 muestras, con una tasa de superposición de 50% o 75%, de modo que la ventana se desliza de 512 o 256 muestras entre cada trama de análisis. Para 44.100 Hz, un tamaño de salto de 512 muestras da una cuantificación temporal de 5,6 ms, lo que es una resolución razonable para distinguir inicios separados por unas pocas decenas de milisegundos.

4.5.1.1 Contenido en Alta Frecuencia.

Para favorecer la detección en banda ancha de un incremento de energía con respecto a otros cambios de energía, tales como la modulación de amplitud, se puede dar a los componentes de alta frecuencia del espectro una ponderación mayor. Masri [1996] propuso una función de contenido de alta frecuencia (HFC), construida mediante la suma de los valores linealmente ponderados de las magnitudes espectrales:

$$D_H[n] = \sum_{k=1}^{N} k|X_k[n]|^2$$

donde Xk [n] es el bin k-esimo de la STFT tomada en el instante n. En esta operación se hace hincapié en los cambios de energía que ocurren en la parte superior del espectro,

sobre todo el incremento de ruido de banda ancha, por lo general asociado con inicios de percusión. Sin embargo, la función es menos eficiente en la identificación de inicios no percusivos como frases de legato, cuerdas frotadas, o flauta, que no presentan esas ráfagas en banda ancha.

4.5.1.2 Diferencia Espectral.

Componentes armónicas que se deslizan de una frecuencia fundamental hasta otra pueden perderse por las funciones de detección de energía y HFC, por ejemplo, cuando sólo se observan pequeños cambios de energía. Otros métodos intentan compensar las deficiencias del HFC en la medición de cambios en el contenido de armónicos de la señal. Uno de estos métodos, conocido como Diferencia Espectral [Foote y Uchi-hashi, 2001], calcula una función de detección basada en la diferencia entre las magnitudes espectrales de dos frames sucesivos de la STFT:

$$D_s[n] = \sum_{k=0}^{N} \Big| \, |X_k[n]|^2 - |X_k[n-1]|^2 \, \Big|.$$

Esta función intenta cuantificar la cantidad de cambio que se encuentra de una trama a otra, en lugar de mediciones frame-by-frame implementadas por sendas funciones de energía y HFC.

4.5.1.3 Desviación de Fase.

Como alternativa, se implementa un enfoque diferente [Bello et al., 2003] consistente en la construcción de una función que mide la inestabilidad temporal de la fase. Onsets con afinación se identificarán por variaciones de fase importantes. El incremento explosivo de energía en inicios de percusión también presentará estas variaciones de fase.

Se espera que una señal estacionaria ha de tener la fase constantemente girando alrededor del círculo unitario. El retardo de fase, la velocidad angular, puede suponerse que es

constante, y su aceleración nula. Los cambios de fase por lo tanto pueden detectarse mirando la fase de aceleración. La función se puede construir mediante la cuantificación de la desviación de fase en cada bin como:

$$\hat{\phi}_k[n] = \text{princarg}\left(\frac{\partial^2 \phi_k[n]}{\partial n^2}\right),$$

ecuación 2.4

donde la fase de los mapas de princarg se encuentran en el rango el rango [-π, π]. Una función de detección de onsets útil se genera como:

$$D_\phi[n] = \sum_{k=0}^{N} |\hat{\phi}_k[n]|.$$

Un inconveniente de esta función es que también pueden producirse cambios de fase importantes en

lugares no relacionados con un cambio musical: componentes ruidosos de la señal por lo general presentan una fase inestable. Aunque esto no puede afectar a los eventos tonales con fuertes componentes armónicos, las grandes variaciones pueden ocurrir como cuando la señal se hace más percusiva y ruidosa.

4.5.1.4 Distancia en el Dominio Complejo.

Con el fin de cuantificar sendos tipos de inicio, de percusión y de tono, la diferencia espectral y los enfoques basados en fase se puede combinar en el dominio complejo [Duxbury et al, 2003.] Para generar una predicción para la trama espectral actual, Xk [n] = | Xk [n] | ejφk [n], donde φk es la función de desviación de fase definida en la ecuación. 2.4. A continuación, mediante la medición de la distancia en el dominio complejo entre el objetivo y la STFT observada se obtiene:

$$D_C[n] = \sum_{k=0}^{N} \left\| \hat{X}_k[n] - X_k[n] \right\|^2 .$$

Esta medida, similar a una distancia euclidiana, pero en el dominio complejo, evalúa la distancia entre la trama actual y la trama predicha a partir de la anterior suponiendo que tanto el desplazamiento de fase y la amplitud son constantes.

4.5.1.5 Distancia Kullback-Liebler.

Se pueden utilizar otras medidas alternativas para evaluar la distancia entre dos vectores espectrales consecutivos. Como estamos viendo incrementos de energía destacados, sin tener en cuenta la disminución, la distancia Kullback-Liebler se puede utilizar para resaltar grandes variaciones e inhibir pequeñas:

$$D_{\mathsf{kl}}[n] = \sum_{k=0}^{N} |X_k[n]| \log \frac{|X_k[n]|}{|X_k[n-1]|}.$$

Esta función acentúa los cambios de amplitud positivos: se generan grandes picos cuando la señal pasa del silencio a un evento, ya que el denominador será mucho menor que el numerador. Una variación de esta función se propone en [Hainsworth y Macleod, 2003], que elimina la ponderación| Xk [n] |, acentuando los cambios de amplitud en la función:

$$D_{\mathsf{mkl}}[n] = \sum_{k=0}^{N} \log \frac{|X_k[n]|}{|X_k[n-1]|}.$$

Para evitar que la función pueda alcanzar valores negativos, lo que aumentaría la complejidad de la selección de picos, y para garantizar que la función se define incluso cuando se encuentra una serie de valores pequeños, podemos modificar aún más la función de la siguiente manera:

$$D'_{\mathsf{kl}}[n] = \sum_{k=0}^{N} \log \left(1 + \frac{|X_k[n]|}{|X_k[n-1]| + \epsilon} \right),$$

donde ε es una constante pequeña, típicamente $\varepsilon = 10^{-6}$. Esta constante está diseñada para evitar grandes variaciones cuando se encuentran niveles de energía muy bajos, y por lo tanto evita grandes picos en la función de detección D '[n] en los eventos offset.

4.5.2 Perfiles de funciones de detección de onsets.

En la Figura 2.2 y la Figura 2.3, se muestran ejemplos de perfiles de detección de onsets obtenidos para dos grabaciones polifónicas. El primer ejemplo (Figura 2.2) es un extracto de una canción brasileña, Azymuth, que contiene un conjunto de viento metal y tambores. La estructura rítmica del extracto aparece claramente definida en el perfil de cada una de las funciones, con picos más agudos o menos fuertes dependiendo de la función. Las notas de viento metal tienden a crear unas pequeñas variaciones en la diferencia espectral en el enfoque basado en fase y en menor medida, en el método en el dominio complejo. Estas variaciones crean picos espurios que el detector de picos debe evitar cuidadosamente a favor de la selección de los picos principales. La función Liebler- Kullback modificada, crea picos afilados en inicios percusivos; en este ejemplo, la función de Liebler-Kullback dará resultados correctos para todos los inicios en el archivo.

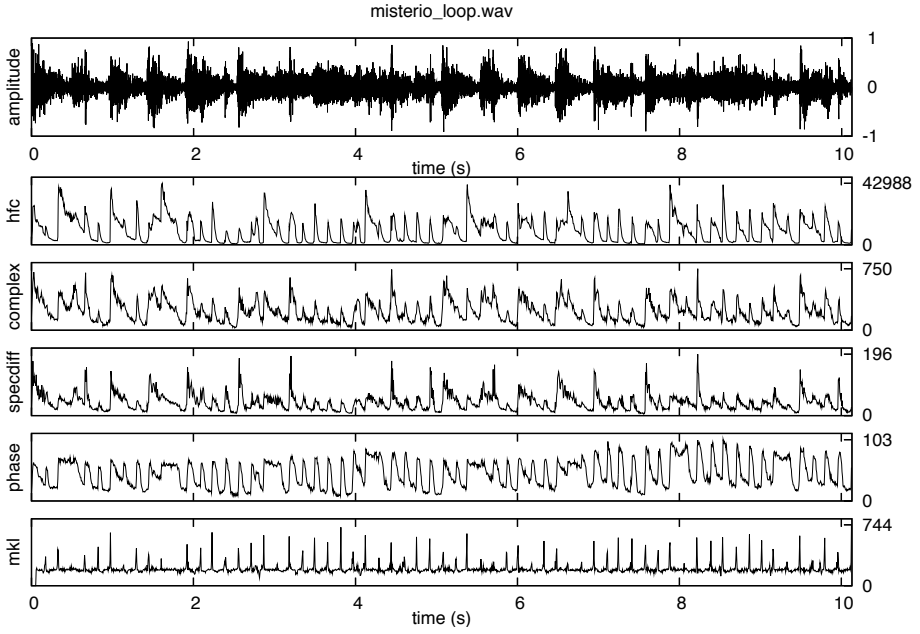


Figure 2.2: Examples of onset detection function profiles: HFC (hfc), Complex domain (complex), spectral difference (specdiff), Phase (phase), Modified Kullback-Liebler (mkl). Sound sample: Misterio, Azymuth

El segundo ejemplo de la Figura 2.3 muestra las primeras medidas de la Quinta Sinfonía de Beethoven. Los violines comienzan con 8 notas fuertes de 0 a 6 s y luego continua una frase de piano, de 6 a 14 s, antes de que el timbal comience a percutir, a partir del segundo 14 hasta el final del archivo. El perfil del HFC permite distinguir claramente los picos más grandes. Sin embargo, los picos en las notas de baja energía tienen una magnitud muy pequeña. Estas diferencias de magnitud tienden a dificultar las operaciones de umbralización y detección de picos. El enfoque Liebler-Kullback no alcanza a ser satisfactorio en la detección de inicios tonales con componentes transitorios débiles. El perfil de la función de detección basado en la fase es la única que contiene todos los picos correspondientes a los inicios de notas reales, y a pesar de la presencia de ruido, esta función permite obtener mejores resultados después de seleccionar los picos relevantes. Las diferencias entre los perfiles obtenidos para las dos grabaciones y la presencia de grandes cambios de amplitud dentro de cada ejemplo, ilustran la dificultad de determinar el mejor algoritmo para seleccionar todos los picos relevantes en las funciones de detección de onsets.

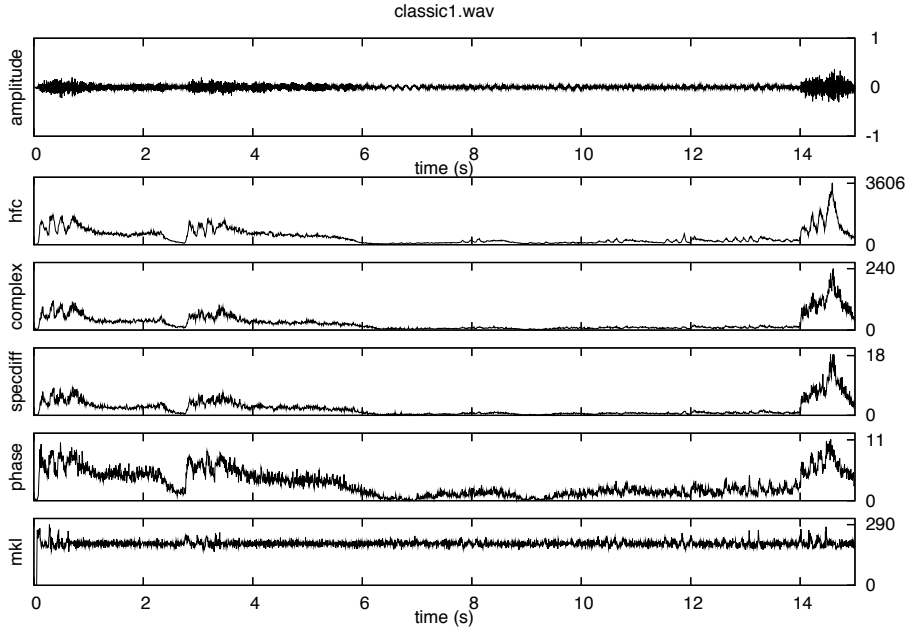


Figure 2.3: Examples of onset detection function profiles: HFC (hfc), Complex domain (complex), spectral difference (specdiff), Phase (phase), Modified Kullback-Liebler (mkl). Sound sample: First measures of the 5^{th} Symphony, Beethoven

4.5.3 Selección temporal de picos de onsets de notas.

La selección final de las ubicaciones de inicio consiste en la identificación de máximos locales en las funciones de detección que corresponden a inicios perceptibles. Dependiendo del contenido de la señal, los picos presentes en la función de detección serán más o menos nítidos y agudos, y pueden ser enmascarados por el ruido, ya sea debido a ruido real en las señales de música o para otros aspectos de la señal, tales como vibrato y modulación de amplitud. Intuitivamente, la caracterización de los tiempos de activación en la función de detección se reduce a una operación de selección de picos: la selección de los máximos locales por encima de un valor umbral dado. Se requieren métodos de selección de picos temporales eficaces para una identificación robusta de tiempos de inicio de la función de detección. En lugar de seleccionar máximos locales, Pucette et al. [1998] propuso seleccionar tiempos de inicio cuando se producen aumentos bruscos de amplitud de la función de detección.

Esta aplicación ha sido probada de manera informal para detectar con precisión inicios de percusión con retardos cortos. Sin embargo, varios onsets de menor energía se descartan en las grabaciones de polifónicas, y eventos con tonalidad se pierden. La detección de estos aumentos es eficiente en ataques abruptos, pero falla en ataques largos en los que el crecimiento de la función de detección es demasiado lento. Se han propuesto enfoques alternativos para la selección de los tiempos de inicio, como por ejemplo, se describen el uso de técnicas de aprendizaje automático para identificar algunas formas características en la función de detección en [Abdallah y Plumbley, 2003, Tindale et al., 2004]. Debido a su complejidad y alto coste computacional, estos métodos son difíciles de aplicar en tiempo real. Para desarrollar una robusta selección de picos se han mostrado en una variedad de funciones de detección [Bello et al., 2005] implementaciones fuera de línea del proceso de selección. En este trabajo el autor revisa algunos de estos enfoques a la Detección de Picos de la función de detección de onsets, e investiga su aplicación en un contexto en tiempo real.

4.5.4 Postprocesado.

Se pueden ejecutar algunos procesos que pueden limitar el número de picos espurios en las funciones de detección antes de buscar los máximos locales. Las operaciones de post-procesado que se aplican a las funciones de detección incluyen el filtrado paso bajo, eliminación de componente continua y normalización [Bello et al., 2005]. El Filtrado Paso Bajo de la función de detección tiene como objetivo reducir el ruido de la señal y reducir al mínimo las detecciones espurias. El filtro puede ser implementado de manera eficiente y causal con un filtro FIR:

$$\tilde{D}[n] = D[n] + \sum_{m=1}^{M} a_m D[n-m].$$

Esta operación reduce el número de picos espurios en la función con un coste adicional mínimo. El Filtrado Paso Bajo está por lo tanto bien adaptado para una aplicación en tiempo real. Para evitar el retraso implicado por el filtro paso bajo, una ventana de la función de detección alrededor de la trama actual se filtra en ambas direcciones, simulando un retraso de fase cero.

Las etapas de normalización y supresión de componente continua y delimitan la función en un rango fijo, típicamente entre 0 y 1. Estos pasos garantizan que la función tiene un perfil determinado independientemente de la amplitud y la naturaleza del sonido, mejorando así el éxito de la operación de umbralización a través de una colección de muestras. Fuera de línea, la normalización y los procesos de eliminación de CC usan la información a partir de un segmento de tiempo grande, tanto antes como después de la trama actual, lo que permite el uso de parámetros fijos para la umbralización. En tiempo real, se puede aproximar mediante el uso de una ventana con bastante tiempo de deslizamiento, lo que aumentaría significativamente el retardo del sistema. Por lo tanto, la eliminación de CC y la normalización no son adecuadas para ser implementadas con retardos muy cortos, y no son aptas para operaciones en tiempo real.

4.5.5 Umbralización dinámica.

Para obtener la secuencia de inicios, los picos de la función de detección post-procesada que corresponden a tiempos de inicio reales deben ser identificados, pero evitando picos espurios. Se pueden observar variaciones de amplitud importantes en las funciones de detección, dependiendo del contenido de la señal, y, en particular, el volumen, como se puede ver en la **figura 2.3**, cuando los timbales entran después del segundo 14. Para compensar los cambios de amplitud pronunciados en el perfil de la función, se utiliza el umbral dinámico: para cada observación en la función de detección, un umbral se calcula sobre la base de un pequeño número de observaciones pasadas y futuras; la amplitud de la observación actual se compara entonces con este umbral. Los métodos para la construcción de un umbral dinámico incluyen un historial de frames [Hainsworth y Macleod, 2003], en el que la amplitud más probable de la función de detección se determina mediante el estudio de población de las observaciones alrededor del instante actual. La mediana de movimiento ha demostrado ser un éxito para reducir el ruido y limitar el número de picos espurios [Rabiner et al., 1975]. Este enfoque se aplicó con éxito en las funciones de detección de onsets, suavizando picos pequeños al tiempo que perfecciona picos de amplitud mayor [Bello et al., 2005]. El filtrado de mediana es también computacionalmente eficiente, ya que la mediana se puede obtener simplemente por ordenar un vector que cuesta significativamente menos que construir un histograma. El umbral dinámico se calcula utilizando el valor de la mediana en un pequeño buffer alrededor de la muestra actual:

$$\delta_t[n] = \lambda \cdot \mathsf{median}(D[n-a], \cdots D[n], \cdots D[n+b]) + \delta,$$

donde la sección D [n - a], · · · D [n], · · · D [n + b] contiene “a” frames espectrales antes de n y “b” frames después de n. El factor de escala λ y el umbral afinado δ son parámetros predefinidos. Los onsets se seleccionan entonces en máximos locales de D [n] - dt [n]. Los buffers utilizados para esta operación por lo general incluyen aproximadamente a + b = 8 frames tomados antes y después de la muestra actual de detección, menos de 100 ms para un sonido a 44100 Hz y tamaño de salto de 512 muestras.

4.5.6 Selección de picos en tiempo real.

Para lograr una sólida selección de máximos pertinentes en un plazo de decisión breve, se propone un enfoque modificado que construye un umbral dinámico basado en una pequeña ventana en torno a la ubicación actual. El umbral dinámico dt [n] está diseñado para permitir la detección de picos en las funciones normalizadas sin componente continua. Para compensar la ausencia del proceso de eliminación de CC y de normalización, se elige una operación de umbralización alternativa. En esta implementación, el umbral dinámico favorece tanto la mediana y la media de una sección de la función de detección, centrada alrededor del marco candidato:

$$\begin{aligned}\tilde{\delta_t}[n] &= \lambda \cdot \mathsf{median}(D[n-a], \cdots D[n], \cdots D[n+b]) \\ &+ \alpha \cdot \mathsf{mean}(D[n-a], \cdots D[n], \cdots D[n+b]) \\ &+ \delta,\end{aligned}$$

ecuación 2.12

donde α es un factor de ponderación positiva. El filtrado de mediana de movimiento se utiliza de una manera similar a la aplicación off line, excepto por que se utiliza un buffer mas corto. El valor de b en la ecuación se reduce al mínimo con el fin de reducir el retardo de la etapa de umbral dinámico. La introducción del valor medio intenta replicar los efectos de la normalización y los procesos de eliminación de CC, sin el uso de una ventana de tiempo, mediante el uso de un valor dinámico para el umbral. Este paso permite que el proceso de selección de picos pueda hacer frente a grandes cambios en la dinámica que se encuentran en señales de música. Los resultados experimentales [Brossier et al, 2004b.] han confirmado que, para valores pequeños de a y b, el umbral modificado es robusto a los cambios dinámicos en la señal.

El proceso de selección de picos en las funciones de detección se realizó usando una ventana móvil de tamaño a = 5 y b = 1 en la ecuación.

Este umbral dinámico modificado puede ser visto como una manera simple para modelar los post-enmascaramientos y enmascaramientos de frecuencia: se selecciona un pico de la

función de detección si su amplitud se encuentra por encima de la amplitud media de las observaciones anteriores. Si se observan grandes amplitudes en varias tramas consecutivas, sólo se seleccionará el primer pico. Aquí hacemos la suposición de que el sistema puede determinar si la trama actual es un inicio o no, dependiendo sólo de una ventana de una trama cercana en el pasado, y sin tener en cuenta los acontecimientos futuros en la señal de audio.

Después de que la función de detección de inicio ha sido post-procesada y se ha calculado un umbral dinámico, el proceso de selección de picos se reduce a la selección de los máximos locales por encima del umbral. La detección de un máximo local implica la comparación de al menos tres observaciones consecutivas, lo que requiere el conocimiento de una observación después del pico. Se define un Onsets como cualquier máximo local en la función de detección de picos:

$$\hat{D}[n] = D[n] - \hat{\delta}_t[n],$$

siendo D [n] una de las funciones de detección definidas y δ [n] se define en la **ecuación. 2.12**. Para reducir el retraso en la selección de picos, sin embargo, se minimiza el impacto en la detección de inicios suaves, seleccionando todos los picos positivos definidos por tres marcos espectrales consecutivos y que se encuentren por encima del umbral dinámico.

4.5.7 Puerta de silencio y pre-enmascarado.

Pruebas de escucha informales han demostrado que se han encontrado un gran número de falsas detecciones en una grabación de vinilo, más alto que en una grabación de CD de la misma pieza, donde el nivel de ruido de fondo es menos prominente [Brossier et al., 2004b]. Las variaciones de amplitud en las zonas de baja energía no pueden ser percibidas como inicios, sin embargo, se observan como picos en las funciones de detección. Para rechazar detecciones falsas en áreas de baja energía, se construye un sencillo detector de envolvente mediante la medición de la energía media de una ventana de la señal. El detector de envolvente actúa como una puerta de silencio, que evita varias detecciones falsas en el fondo y el ruido de cuantificación, donde los onsets son más propensos a ser producidos por el ruido de fondo. Por otra parte, una medida de la intensidad de la señal es útil para detectar tiempos de desplazamiento: un marco con una energía media por debajo

de un umbral dado después de un marco con una energía media por encima de este umbral indica un desplazamiento. El umbral de la puerta de silencio debería ser elegido para evitar detecciones falsas, no sólo entre las canciones, sino también durante los períodos de silencio cortos dentro de algunas canciones. Mediante el uso de la puerta de silencio para descartar inicios detectados en las regiones de baja energía, se podrían alcanzar mejoras significativas en la exactitud de la detección.

Debido a que el umbral dinámico utiliza casi exclusivamente la información pasada, no tenemos medios para detectar cuando un pico en la función de detección será seguido en breve por otro pico más grande. El sistema es por lo tanto propensos a causar detecciones duplicadas. Con el uso de un intervalo mínimo entre la aparición, podemos asegurar que dos inicios consecutivos no serán detectados en menos de este intervalo. El parámetro para el intervalo mínimo entre la aparición controla el intervalo de tiempo más corto después del cual se puede detectar un nuevo onset. Obviamente, si se establece un intervalo mínimo entre onsets se reduce el número de falsos positivos, provocados por ejemplo por la amplitud o modulación de frecuencia. Sin embargo, el valor del intervalo entre la aparición mínima debe ser lo suficientemente corto para identificar sucesiones rápidas de onsets. Se ha medido experimentalmente que el uso de un retraso de tiempo de 20 ms a 30 ms es lo suficientemente largo para evitar varios falsos positivos, sin afectar a la precisión global.

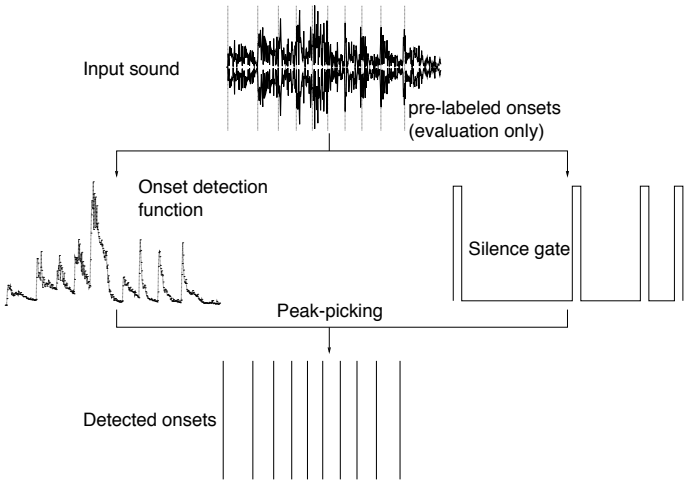


Figure 2.4: Overview of the segmentation process: the audio signal is reduced to an onset detection function at a lower sampling rate, and peaks are searched for in this function; a silence gate is used in parallel to prevent spurious detections in areas of low energy.

4.5.8 Implementación del sistema.

La **Figura 2.5** da una visión general del proceso que utilizan en Aubio para la extracción de tiempos de inicio. La señal de audio se reduce primero a una Función de Detección de inicio a una tasa de muestreo más baja. A continuación, se realiza una Selección de Picos en la función de detección para obtener una secuencia de tiempos de inicio. Esta secuencia se combina con la salida de un detector de silencio para definir los pares de onsets/offsets que definen los límites de los objetos de nota.

Los itinerarios necesarios para la detección se muestran tanto para la aplicación en línea como fuera de línea. En ambos casos, el filtrado paso bajo y mediana móvil se utilizan para eliminar el ruido y la fluctuación de fase y seguir las variaciones de amplitud. En el proceso de selección de picos fuera de línea, se utilizan supresores de CC y normalización para obtener perfiles de función de detección uniformes a través de una colección de muestras de sonido. En línea, la media móvil tiene por objeto sustituir estos dos pasos.

Después de la transformación de Vocoder de Fase y la obtención de la función de detección, el tiempo de inicio detectado se retrasa unos pocos frames pasado el tiempo de

ataque real de la señal. El retardo teórico es de (3 + b) · tamaño de salto de ventana/ frecuencia de muestreo, donde se requieren tres frames para detectar un pico, b para el paso de umbral dinámico. Para una tasa de muestreo de 44100 Hz y un tamaño de salto de 256 muestras, utilizando b = 1 para calcular el umbral dinámico, el retardo de sistema esperado es de 23,2 ms y puede reducirse aún más mediante el uso de tamaños de salto más cortos.

Dicho retardo es aceptable para un tiempo de ataque perceptual, y los inicios extraídos en tiempo real se pueden utilizar para desencadenar eventos de audio o visuales sin demora perceptible. Para la edición de audio, las operaciones de cortar y pegar y otros usos de corte de señal, la ubicación de la detección debe ser tan precisa a la muestra tanto como sea posible. Las diferentes funciones de detección de inicio tienden a alcanzar su punto máximo en el cambio máximo en el ataque, el pico se retrasa aún más por la etapa de post-procesamiento. Se requiere la eliminación apropiada del retardo del sistema para una localización precisa de los tiempos de inicio. Para reducir el clicking y los artefactos producidos por saltos de fase, que se obtendrían por concatenación de cortes individuales, es preferible la selección de un punto de cruce por cero en la forma de onda. A partir de este mínimo local, asegurando que el ataque del siguiente corte se conserva, se busca el cruce por cero más cercano para seleccionar la mejor ubicación de edición.

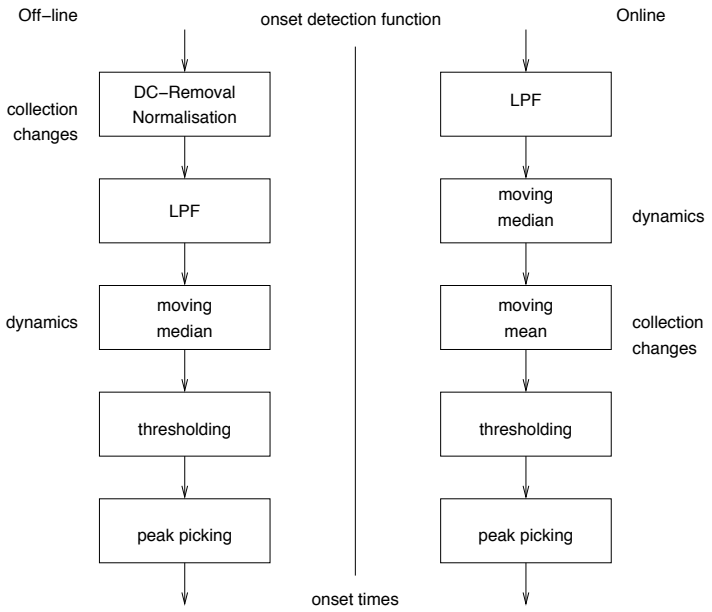


Figure 2.5: Comparison of off-line and online peak picking methods. Off-line, DC-removal and normalisation are used to cope with loudness variations across the database; the dynamic threshold modified for real time operation (Eq. 2.12) uses a moving mean to cope with loudness changes.

4.5.9 Resultados.

A partir de una base de datos de canciones etiquetadas a mano, se comparan los resultados de tiempos de onsets arrojados por el sistema.

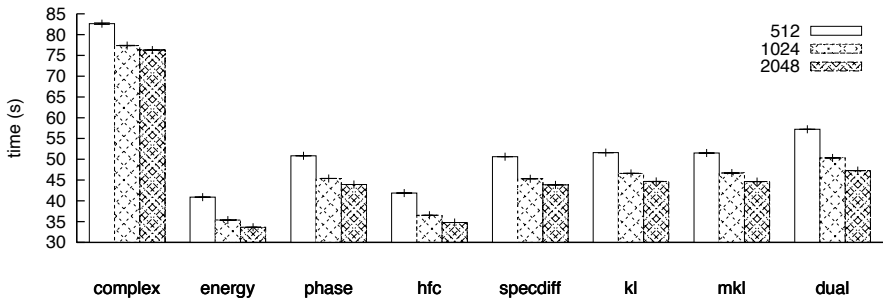


Figure 2.11: Computation times in seconds for different onset algorithms on the Mirex 2005 database (Table 2.1, approx. 23 min) at window sizes 512, 1024 and 2048, with 50% overlap. Tests were run on an Apple iBook G4 1.0 GHz running Debian GNU/Linux Etch.

Algorithm	% F	% P	% R	GD	FP	FN	M	D	Dist	aDist	Run
Lacoste & Eck 2	80.07	79.27	83.70	26.82	6.05	4.85	0.65	0.17	0.00613	0.0115	4713
Lacoste & Eck 1	78.35	77.69	83.27	26.55	7.91	5.12	0.62	0.19	0.00572	0.0115	1022
Ricard, J.	74.80	81.36	73.70	23.97	5.18	7.70	0.63	0.01	0.00593	0.0138	154
Brossier, P.	74.72	74.07	81.95	25.81	10.71	5.86	0.62	0.15	-0.00384	0.0111	50
Röbel, A. 2	74.64	83.93	71.00	22.62	3.46	9.05	0.51	0.52	0.00380	0.0084	159
Collins, N.	72.10	87.96	68.26	21.27	2.13	10.40	0.52	0.12	-0.00120	0.0069	12
Röbel, A. 1	69.57	79.16	68.60	21.40	5.05	10.27	0.48	0.88	0.00525	0.0087	158
Klapuri et al.	58.92	60.01	61.62	19.41	15.08	12.25	0.73	0.18	-0.02209	0.0276	56
West, K.	48.77	48.50	56.29	18.46	24.05	13.21	0.46	0.00	-0.00499	0.0138	179

Table 2.2: Overview of results of the MIREX 2005 Audio Onset Detection Contest [MIREX, 2005b]: overall average F-measure (F), precision (P) and recall (R); average number of correct detection (GD), false positives (FP), false negatives (FN), merged (M) and doubled (D); mean distance (Dist) and absolute mean distance (aDist) to hand labelled onsets; average runtime per file (Run).

Algorithm	Thresh.	% F	% P	% R	GD	FP	FN	M	D	Dist	aDist
complex	0.4	74.1	81.1	68.2	22.5	5.22	10.4	0.78	1.14	0.00439	0.00906
energy	0.8	63.8	70.9	57.9	19.1	7.82	13.8	0.67	1.00	0.00875	0.01286
phase	0.3	74.4	79.0	70.2	23.1	6.13	9.79	0.81	1.52	0.00133	0.00872
hfc	0.3	76.6	80.7	72.9	24.0	5.72	8.91	0.98	2.01	0.00656	0.01131
specdiff	0.3	75.0	77.0	73.2	24.1	7.17	8.82	0.88	1.58	0.00643	0.01095
kl	0.4	73.1	79.4	67.7	22.3	5.78	10.6	0.86	1.27	0.00368	0.00850
mkl	0.2	67.7	75.7	61.3	20.2	6.49	12.7	1.24	3.95	-0.00005	0.01107
dual	0.3	76.1	76.9	75.3	24.8	7.44	8.12	1.18	2.92	0.00355	0.01061

Table 2.3: Onset detection results obtained after training with our aubio real-time implementation on database MIREX 2005. The peak-picking threshold is indicated in the second column. Following column legends are identical to the ones in Table 2.2.

En este trabajo el autor propone modificaciones simples basadas en la percepción a los algoritmos de Detección de Picos existentes, y los experimentos en las bases de datos han demostrado que el impacto del algoritmo de peak picking se limita a unos pocos timbres problemáticos. La aplicación causal abre el camino a nuevas aplicaciones, con el remuestreo en vivo y la construcción de la al vuelo de los segmentos anotados. Por otra parte la extracción de onsets es rápida y robusta y puede mejorar significativamente la velocidad de los sistemas que requieren segmentación temporal.

Se ha presentado un marco de trabajo completo para la evaluación de la ejecución de estas funciones. La evaluación en las bases de datos mostró que varios métodos podrían lograr una extracción precisa de los pulsos de inicio sin ajuste de los parámetros. Con uso de un solo parámetro, una combinación perfecta se puede obtener en más de 90% de los ejemplos de sonido. El marco de evaluación ha puesto de relieve los beneficios de la computación simultánea de dos funciones diferentes. Como todas las funciones utilizan el mismo marco espectral, calculando varias funciones de detección se obtiene un ahorro computacional. Este modo dual es la que se propone como la configuración predeterminada.

5. Metodología y herramientas para la evaluación de un sistema de detección automática de onsets en música.

A partir de una batería de pruebas, compuesta por una selección de piezas musicales reales polifónicas y multitímbricas etiquetadas de forma manual, podemos evaluar nuestro sistema de tres formas diferentes:

- Señales de audio en las que podemos escuchar sonidos sintetizados en los instantes de tiempo donde un onset es detectado.
- Inspección visual de la forma de onda, con los instantes de onsets señalados sobre ella, mientras escuchamos el contenido de la señal.
- Evaluación automática del sistema, por medio de un algoritmo de evaluación de detección automática de onsets.

5.1. Herramientas.

Para etiquetar los onsets reales en los ficheros de audio y establecer el "groundtruth" para nuestro banco de pruebas, se han etiquetado los onsets mediante dos procesos:

- Etiquetado manual de señales de audio de grabaciones reales mediante inspección auditiva y visual de la forma de onda de la señal.

- Obteniendo los inicios de nota de las instrucciones "note on" de ficheros MIDI posteriormente sintetizados.

- En el primero de los casos se ha usado Sonic Visualizer, una aplicación para visualizar y analizar el contenido de ficheros de audio, que resulta especialmente interesante a musicólogos, catalogadores, investigadores en procesado de señal y cualquiera que necesite conocer de manera sencilla los entresijos de un archivo de audio digital. Se trata de un software libre distribuido bajo licencia GNU General Public License y desarrollado por el Centro para Música Digital Queen Mary de la Universidad de Londres. Para la evaluación automática del sistema se ha implementado un algoritmo de evaluación que será descrito mas adelante ene esta memoria.

- El en segundo para la conversión de MIDI a etiquetas de onsets se ha usado la herramienta smf2txt, una herramienta de linea de comandos que nos permite extraer diversa información de un fichero MIDI y exportarla a un fichero de texto en diferentes formatos. En este trabajo se usa para convertir las instrucciones MIDI note on en lineas de un archivo de texto plano que contienen el tiempo de comienzo de la nota en segundos. txt2smf es la herramienta reciproca a esta. Ambas han sido desarrolladas en el Dept. Lenguajes y Sistemas Informáticos de la Universidad de Alicante por Pedro j. Ponce de León, David Rizo y José Manuel Iñesta, "A command line toolchain for MIDI file processing".

- Para la síntesis de audio de ficheros MIDI se ha utilizado el sintetizador General MIDI del secuenciador de apple Logic Pro.

- Se ha recurrido a hojas de cálculo para acondicionar los ficheros de texto provenientes de MIDI a las condiciones psicoacústicas establecidas, así como para ciertos análisis de los ficheros ground truth y cálculos diversos.

- El etiquetado automático de los archivos de audio para las diferentes piezas de nuestra base de datos en los diferentes algoritmos Vamp se ha implementado mediante Sonic Annotator, que es un complemento de Sonic Visualizer que permite gestionar varios archivos de forma simultánea.

- La base de datos de archivos de audio proviene de la RWC Music Database, concretamente las piezas 1-10 de la colección RWC-MDB-J-2011-M01. RWC es una de las más famosas bases de datos de música a gran escala compilada específicamente para fines de investigación.

- El evaluador automático y el sistema de propagación de corrección de errores se han desarrollado mediante la programación de aplicaciones en C.

5.2 Etiquetado manual de onsets en señales musicales.

Para poder utilizar el sistema de evaluación automática necesitamos además de las señales a analizar, los momentos de onsets reales o de referencia de cada una de ellas. Estos tiempos de referencia se obtienen mediante el etiquetado manual de onsets.

El etiquetado manual de onsets en señales musicales se refiere al establecimiento de los tiempos exactos de comienzos de eventos por un oyente humano. Esta actividad genera dos cuestiones que no están cerradas, y que son base de distintos trabajos y artículos. Estas dos cuestiones son:

- Definición exacta de onset.
- Metodología de evaluación de una base de datos etiquetada manualmente.

5.2.1. Caracterización de un onset. Particularidades de los onsets en señales musicales.

Antes de explicar el proceso de etiquetado de canciones, se intentará definir exactamente qué es un onset. La definición usada anteriormente dice que "es el instante de tiempo donde comienza una nota", sin embargo, esta definición es muy ambigua. En primer lugar, la totalidad de las señales que vamos a analizar son grabadas. Esto implica que el instante exacto donde se produce un onset no tiene por qué ser visible/audible en la señal con la que estamos trabajando. Por otro lado, durante la grabación de una señal pueden ocurrir eventos descontrolados y no deseados, como por ejemplo, el ruido producido por los instrumentos de la familia viento/madera al recibir la respiración del intérprete, el cual es audible si se presta atención pero que no tiene significado estético ni musical. De hecho, algunos investigadores se preguntan si a la hora de etiquetar una señal musical sería importante tener en cuenta estos eventos. Además, toda esta problemática se agrava en el momento en se deja de pensar en una nota como un evento aislado. Cuando se encuentra una secuencia musical como un solo, la determinación del comienzo de las notas implicadas se convierte en una cuestión mucho más compleja, mayor cuanto mayor es la velocidad de la interpretación. Por otro lado en el caso de instrumentos monofónicos, no se pueden obviar los factores del entorno en el momento de la grabación, ya que por ejemplo, un tiempo de reverberación alto podría aumentar el tiempo de relajación de una nota prolongándolo mas allá del instante de inicio de la siguiente, quedando así enmascarado el instante exacto de su comienzo. En el caso de instrumentos polifónicos se suman a este otros fenómenos que pueden enmascarar el instante de comienzo de una nota. En el caso de la ejecución de un acorde, si éste es tocado al unísono se considera un solo onset, pero crea incertidumbre cuando alguna de las notas del acorde suena un poco desplazada (por ejemplo, en una guitarra transcurre un tiempo desde la excitación de la sexta cuerda hasta la excitación de la primera). Por otro lado, en los instrumentos de cuerda frotada es muy difícil discernir el comienzo de una nota cuando la anterior ha sido ejecutada en una cuerda diferente. Todos estos efectos negativos se ven multiplicados cuando se habla de música polifónica y multitímbrica, múltiples instrumentos en múltiples pistas a diferente volumen y en diferente posición espacial. Por todos estos motivos, se considera la detección manual de onsets una cuestión compleja y relativamente subjetiva.

5.2.2. Metodología de etiquetado manual de onsets.

Como ya se ha indicado, para poder evaluar un sistema de detección automática de onsets es necesario comparar los resultados del sistema con unos tiempos de localización de referencia, llamados onsets reales, que desafortunadamente no existen como tal excepto en un limitado número de casos como, por ejemplo, en determinada música sintetizada. Por este motivo surge la necesidad de elaborar estos datos de referencia de forma manual. En la literatura se puede observar que las mismas señales de prueba, etiquetadas por diferentes individuos, producen instantes de referencia diferentes, mostrando una gran dependencia del método utilizado para etiquetar, de las características de la señal, y del propio oyente. De estos estudios se concluye que la evaluación de un sistema de detección automática de onsets en señales musicales es una cuestión compleja y nada trivial.

El etiquetado manual de onsets es una tarea laboriosa, que requiere tiempo y concentración por parte del sujeto que la realiza. Normalmente en el etiquetado de onsets se utilizan tres métodos:

- Representación gráfica de la señal: Es un método muy eficiente y rápido si se trata de señales percusivas de gran amplitud, ya que muestra claramente una alteración en la forma de onda cuando se produce una nota. Para el resto de señales se suele usar como método de apoyo.

- Espectrograma: Puede ser usado como primera aproximación. Debido a la necesidad de tomar una ventana de gran numero de muestras al hacer la FFT para poder tener una buena resolución espectral, este método no es muy preciso. Sin embargo, es de gran ayuda a la hora de localizar un onset en un rango determinado.

- Escuchando pequeñas secciones de la señal: Se escucha la señal en secciones de unos pocos segundos y se determina el número de onsets y la posición de los mismos dentro del pasaje. En algunos casos reducir la velocidad de reproducción, siempre

que este proceso no altere el tono de la interpretación, puede facilitar el proceso, aunque a veces puedan aparecer artefactos sonoros que podemos confundir con un onsets en notas alargadas o perder algunos inicios sutiles.

Además de estos métodos existen otras alternativas como diversas representaciones de la señal, por ejemplo, transformadas wavelets, representación de la fase, …, en donde se pueden observar los cambios en algunas de sus características al comenzar una nota.

En este proyecto se ha usado una técnica híbrida de etiquetado manual: visualizar una pequeña sección de la forma de onda y escuchar su audio correspondiente, para fijar sobre ésta los instantes de onsets. Para esta tarea se ha usado Sonic Visualiser.

5.2.3. Elaboración de una base de datos para evaluar el sistema.

Para evaluar el sistema se ha preparado una batería de pruebas consistente en 15 canciones con sus respectivos archivos de etiquetas, realizados a mano y mediante la extracción de datos MIDI del modo descrito anteriormente. Dentro de esta selección se ha intentado incluir diferentes escalas, velocidades en la ejecución de las notas, grandes rangos dinámicos, notas percusivas, notas mas suaves, etc, con la condición de que las piezas sean polifónicas y monotímbricas, aunque puede haber mas de un instrumento a la vez siempre que se trate del mismo tipo. Esta condición es debida a que el sistema de detección de onsets basado en filtros de un doceavo de octava de la Universidad de Alicante, sobre el cual se estudiarán los efectos de un algoritmo de propagación de interacciones con usuarios, es un subsistema de un sistema principal de transcripción de audio a MIDI.

Un tipo de música que responde a estas características es la música Jazz interpretada en piano o guitarra, por lo que nuestra base de datos serán las 5 primeras piezas de grabaciones reales de la colección Jazz Music de la RWC Music Database, RWC-MDB-J-2001-M01 y las 10 primeras piezas de esta misma colección sintetizadas desde MIDI. La RWC Database consta de cientos de canciones de diferentes estilos, disponibles cada una en dos formatos: grabaciones reales y ficheros MIDI. En un principio, para este trabajo se recurrió a esta base de datos con el fin de poder extraer la información de los tiempos de

onsets directamente desde los ficheros MIDI, para después analizar en el sistema las grabaciones reales y comparar los resultados. Para esta tarea se ha usado "smf2txt", desarrollado por el DSLI (Departamento de Lenguajes y Sistemas Informaticos) de la de la Universidad de Alicante. Mediante este software se han extraido a un fichero de texto los eventos MIDI "nota on", que son los que marcan el comienzo de una nota en fichero MIDI. Este procedimiento presenta el problema de que el fichero de texto contiene todos los eventos de comienzo de nota de todos los instrumentos implicados en la síntesis de una canción. Es decir para melodías polifónicas y multitímbricas encontramos inicios de nota que comparten el mismo tiempo o muy cercanos. Teniendo en cuenta que el oído humano solo distingue sonidos separados en el tiempo menos 50 ms y que para un instante de tiempo solo se puede producir un onsets, esta información no resultaba útil. Además la información en el fichero de texto aparece desplazada respecto a la señal original. Para solucionar esto se ha manipulado el fichero de texto mediante hojas de cálculo para fijar conjuntos de onsets no distantes más de 30 ms en único onsets de valor de tiempo promedio. Debido a la asincronía del fichero de audio con el MIDI, se ha optado por etiquetar canciones de forma manual y a partir del MIDI, y realizar su evaluación por separado. De esta forma construimos una base de datos más pequeña, pero de total fiabilidad, se han etiquetado para 5 grabaciones reales los archivos de audio de forma manual y se ha sintetizado el archivo MIDI correspondiente a las otras 10 piezas de las cuales se han extraído y acondicionado los instante de onsets a partir de los "note on" de los ficheros MIDI.

Estas 15 piezas, suman un total de 53 minutos y 26 segundos de audio y presentan 11.553 onsets etiquetados como groundtruth.

Nº	Procedencia	Titulo - interprete	Duración	Onsets
1	*RWC-MDB-J-2001-M01*	Jive - Makoto Nakamura	3:22	857
2	*RWC-MDB-J-2001-M01*	For Two - Makoto Nakamura	6:15	844
3	*RWC-MDB-J-2001-M01*	Lounge Away - Takao Nagai	2:38	701
4	*RWC-MDB-J-2001-M01*	Crescent Serenade - Makoto Nakamura	2:47	993
5	*RWC-MDB-J-2001-M01*	Abyss - Takao Nagai	3:08	689
6	*RWC*	Jive (Piano Solo MIDI)	3:23	860
7	*RWC*	For Two (Piano Solo MIDI)	6:17	868
8	*RWC*	Lounge Away (Piano Solo MIDI)	2:43	692
9	*RWC*	Crescent Serenade (Piano Solo MIDI)	2:46	1.070
10	*RWC*	Abyss (Piano Solo MIDI)	3:04	710
11	*RWC*	Jive (Guitar Solo MIDI)	2:39	691
12	*RWC*	For Two (Guitar Solo MIDI)	3:36	819
13	*RWC*	Lounge Away (Guitar Solo MIDI)	3:39	591
14	*RWC*	Crescent Serenade (Guitar Solo MIDI)	3:40	596
15	*RWC*	Abyss (Guitar Solo MIDI)	3:29	572
Total			53:26	11.553

5.3. Evaluación de sistemas de detección de onsets.

Además de la evaluación visual y auditiva, se hará uso de un evaluador automático implementado en C, para realizar una evaluación del sistema. La aplicación carga dos ficheros de texto, uno con el groundtruth de instantes de onsets reales y otro con los instantes de onsets detectados de forma automática por un sistema de detección, ambos en segundos. El sistema evaluará estos datos según la metodología de evaluación.

5.3.1. Sonido sintetizado en los instantes de onsets detectados.

Este método se utiliza como ayuda en el proceso de desarrollo y calibración del sistema. Al poder escuchar la señal original como referencia audible, se puede apreciar cuándo se producen fallos en la detección como falsos negativos, falsos positivos, o desplazamiento en la posición temporal de los onsets. Además, permite comprobar de manera directa como responde el sistema a los distintos tipos de señales analizadas, rangos dinámicos, volúmenes, velocidad en la ejecución de la pieza, etc. De esta forma, al detectar un fallo en la detección es posible variar los valores de los parámetros del sistema y comprobar mediante el resultado obtenido si éste se ha solventado.

5.3.2. Inspección de la forma de onda con marcas visuales en los instantes de onsets detectados.

Al igual que el método de síntesis de sonido en los instantes de onset, éste método se utiliza como ayuda en el desarrollo del sistema y no para presentar resultados. La comprobación de los instantes de onsets se realiza mediante inspección audio-visual de la forma de onda, constatando en pequeñas secciones de ésta el correcto posicionamiento de las marcas generadas por el sistema de detección. Esta técnica requiere una gran concentración por parte de la persona que realiza la comprobación, que además debe ser un experto oyente. Esta técnica puede ser combinada con la anterior de forma que al presentarse además de la señal original el sonido de los clics, los instantes de detección estarán mucho mas marcados sobre la forma de onda.

5.3.3. Metodología de evaluación automática.

El método de síntesis de sonido en los instantes de onsets y el método de inspección visual y auditiva de la forma de onda presentan la particularidad de ser cuasi-subjetivos, por lo que no se usaran para mostrar resultados. La finalidad de estos métodos es usarlos como ayuda para detectar fallos y establecer parámetros en el sistema. Además, ofrecen una salida del sistema que no se limita a dar información sobre el mismo, si no que presentan un resultado audible/visible que se puede interpretar con mayor facilidad. El método de evaluación automática presenta un problema que suele originar gran controversia y debate a su alrededor: el etiquetado manual de señales musicales, que ya se trato en esta memoria.

Antes de explicar como se han evaluado los sistemas de detección automática, definiremos algunos conceptos necesarios para su comprensión:

- Falso negativo: Se trata de un onset que está presente en la señal original pero que no ha sido detectado por el sistema.

- Falso positivo: Se trata de un onset que no está presente en la señal original pero que ha sido detectado por el sistema.

- Onset correcto: Se trata de un tiempo de onset detectado por el sistema que coincide con uno de los onsets de referencia (onsets reales), o se encuentra dentro de un margen de error temporal muy pequeño respecto a éste.

- Onset real o de referencia: Es el tiempo exacto en el que sucede un onset. La definición de onset real presenta algunos problemas que serán tratados más adelante en este documento.

- Precisión: Porcentaje de onsets detectados que son correctos.

- Recall: Porcentaje de onsets verdaderos respecto a los onsets detectados por el sistema.

5.3.3.1 Evaluación automática del sistema.

El sistema compara los tiempos de referencia de los onsets reales con los de salida del sistema de detección automática, y determina el número de detecciones totales, el número de onsets falsos positivos y el número onsets falsos negativos. Con todos estos datos, el porcentaje de acierto del sistema se calcula como:

$$\Pr ecision = \frac{Total - FalsosNegativos - FasosPositivos}{Total} \times 100\%$$

Para determinar el si una detección es correcta, se debe fijar una precisión temporal. En la mayoría de sistemas de detección de onsets se considera que un tiempo es correcto si se encuentra dentro de una ventana temporal de 50ms respecto a la posición del onset de referencia. En este trabajo se tomará este valor de tolerancia para evaluar el sistema.

Los resultados de este evaluador se dan en términos de Precision, Recall, número de falsos negativos y número de falsos positivos.

6. Evaluación de los sistemas en estudio.

Con el fin de evaluar los diferentes sistemas de detección automática de onsets a los que se ha tenido acceso desde plug-in´s Vamp en Sonic Visualizaer, se analizara en un banco de pruebas sobre cada uno de los sistemas descritos la base de datos de onsets manuales construida, obteniendo de esta manera diversos parámetros de medición de la detección. Dicha evaluación se llevará a cabo mediante un software de evaluación automática, a partir de la información de las etiquetas de onsets reales contra los onsets etiquetados de forma automática por los distintos sistemas.

6.1. Evaluación y resultados.

Para evaluar los distintos sistemas de detección automática de onsets que se han presentado en este trabajo, se ha realizado la detección de cada uno de ellos en cada una de las 15 señales que componen la base de datos de onsets reales que se ha elaborado. Para cada par señal/sistema se han medido los siguientes parámetros, descritos en el apartado anterior de esta memoria:

- OK, número de onsets correctos.
- FP, número de onsets falsos positivos.
- FN, número de onsets falsos negativos
- D, número de onsets doblados.
- M, número de onsets mezclados.
- MR, ratio de onsets mezclados.
- DR, ratio de onsets doblados.
- MD, media de desviación respecto a groundtruth.
- P, precisión.
- R, recall
- F, medida global

Nº, es el numero de señal según tabla 5.2.3. y P es cada parámetro de medida. Los diferentes sistemas utilizados vienen indicados como:

- UA, Universidad de Alicante.
    - Algoritmo basado en bancos de filtros de un doceavo de octava.
- QM, Centro de Música Digital Queen Mary de la Univeridad de Londres.
    - QMAW, Algoritmo “Adaptative Whitening”.
    - QMCD, Algoritmo “Complex Domain”.
- A, Aubio.
    - ACDD, Algoritmo “Complex Domain Distance”.
    - AHQ, Algoritmo “Hihg Frequency”.
    - AKL, Algoritmo distancia “Kullback-Liebler”.
    - APD, Algoritmo “Phase Deviation”.
    - ASD, algoritmo “Spectral Difference”.

Los valores de sistema usados para realizar las detecciones en los plug-in Vamp de Sonic Visualizer han sido los siguientes:

- UA:
  - Sensitivity = 0,18
  - Audio Frames per block = 4096
  - Window increment = 2048
- QMCD:
  - Program = General purpose
  - Function type = Complex Domain
  - Sensitivity = 50%
  - Adaptative Whitening = No
  - Window size = 1024
  - Window increment = 512
  - Window sahpe = Hann
- QMAW:
  - Program = “”
  - Function type = Complex Domain
  - Sensitivity = 50%
  - Adaptative Whitening = Yes
  - Window size = 1024
  - Window increment = 512
  - Window sahpe = Hann
- ACDD:
  - Function type = Complex Domain
  - Peak Picker Threshold = 0,3
  - Silence Threshold = -70,0 dB
  - Minimum inter-onset interval = 4,0 ms
  - Audio frames per block = 1024
  - Window increment = 512
- AHF:
  - Function type = High-Frequency Content
  - Peak Picker Threshold = 0,3

    - Silence Threshold = -70,0 dB
    - Minimum inter-onset interval = 4,0 ms
    - Audio frames per block = 1024
    - Window increment = 512
- AKL:
    - Function type = Kullback-Liebler
    - Peak Picker Threshold = 0,3
    - Silence Threshold = -70,0 dB
    - Minimum inter-onset interval = 4,0 ms
    - Audio frames per block = 1024
    - Window increment = 512
- APD:
    - Function type = Phase Deviation
    - Peak Picker Threshold = 0,3
    - Silence Threshold = -70,0 dB
    - Minimum inter-onset interval = 4,0 ms
    - Audio frames per block = 1024
    - Window increment = 512
- ASD:
    - Function type = Spectral Difference
    - Peak Picker Threshold = 0,3
    - Silence Threshold = -70,0 dB
    - Minimum inter-onset interval = 4,0 ms
    - Audio frames per block = 1024
    - Window increment = 512

En la siguiente tabla se muestran los resultados obtenidos en el banco de pruebas realizado.

Nº	P	UA	QMAW	QMCD	ACDD	AHQ	AKL	APD	ASD
1	OK	718	571	787	699	515	617	638	620
	FP	37	63	36	15	1	5	15	11
	FN	139	286	70	158	342	240	219	237
	D	2	3	2	4	0	1	0	3
	M	2	3	2	4	0	1	0	3
	MR	1.43885	1.04895	2.85714	2.53165	0	0.416667	0	1.26582
	DR	5.40541	4.7619	5.55556	26.6667	0	20	0	27.2727
	MD	0.0197046	-0.0021756	0.0003835	-4.77e-10	-0.00018035	-0.007602	-0.00522267	-0.00159169
	P	0.950993	0.900631	0.956258	0.978992	0.998062	0.991961	0.977029	0.982567
	R	0.837806	0.666278	0.91832	0.815636	0.600933	0.719953	0.744457	0.723454
	F	0.890819	0.765929	0.936905	0.889879	0.750182	0.834347	0.845033	0.833333
2	OK	779	631	773	739	554	669	704	701
	FP	19	99	246	27	3	22	88	23
	FN	65	213	71	105	290	175	140	143
	D	0	3	25	5	2	4	3	3
	M	0	3	25	5	2	4	3	3
	MR	0	1.40845	35.2113	4.7619	0.689655	2.28571	2.14286	2.0979
	DR	0	3.0303	10.1626	18.5185	66.6667	18.1818	3.40909	13.0435
	MD	-0.0000045	-0.0146448	-0.0174769	-0.0200017	-0.0185943	-0.0251295	-0.0241252	-0.0205078
	P	0.97619	0.864384	0.758587	0.964752	0.994614	0.968162	0.888889	0.968232
	R	0.922986	0.74763	0.915877	0.875592	0.656398	0.792654	0.834123	0.830569
	F	0.948843	0.801779	0.829844	0.918012	0.790864	0.871661	0.860636	0.894133
3	OK	583	485	603	550	415	492	505	489
	FP	38	19	64	15	3	17	16	15
	FN	118	216	98	151	286	209	196	212
	D	3	0	1	3	0	3	2	0
	M	3	0	1	3	0	3	2	0
	MR	2.54237	0	1.02041	1.98676	0	1.43541	1.02041	0
	DR	7.89474	0	1.5625	20	0	17.6471	12.5	0
	MD	0.0179896	-0.0018009	-0.0021617	-0.0001567	-0.00078463	-0.00741349	-0.00590907	-0.00193045
	P	0.938808	0.962302	0.904048	0.973451	0.992823	0.966601	0.96929	0.970238
	R	0.831669	0.691869	0.8602	0.784593	0.592011	0.701854	0.720399	0.697575
	F	0.881997	0.804979	0.881579	0.868878	0.741734	0.813223	0.826514	0.811618
4	OK	785	568	835	590	418	510	508	499
	FP	45	28	95	2	1	8	9	4
	FN	208	425	158	403	575	483	485	494
	D	1	0	5	1	1	1	0	1
	M	1	0	5	1	1	1	0	1
	MR	0.480769	0	3.16456	0.248139	0.173913	0.207039	0	0.202429
	DR	2.22222	0	5.26316	50	100	12.5	0	25
	MD	0.0185019	-0.003895	-0.0042297	-0.000440	-0.000982326	-0.00577093	-0.00618646	-0.00168768
	P	0.945783	0.95302	0.897849	0.996622	0.997613	0.984556	0.982592	0.992048
	R	0.790534	0.572004	0.840886	0.594159	0.420947	0.513595	0.511581	0.502518
	F	0.861218	0.714915	0.868435	0.744479	0.592068	0.67505	0.672848	0.667112
5	OK	583	466	615	473	341	405	444	389
	FP	14	77	111	7	3	7	23	10
	FN	106	223	74	216	348	284	245	300
	D	0	3	9	2	1	0	0	1
	M	0	3	9	2	1	0	0	1
	MR	0	1.34529	12.1622	0.925926	0.287356	0	0	0.333333
	DR	0	3.8961	8.10811	28.5714	33.3333	0	0	10
	MD	0.0057394	-0.016365	-0.0150526	-0.015308	-0.0125971	-0.018137	-0.0198669	-0.0148872
	P	0.976549	0.858195	0.847107	0.985417	0.991279	0.98301	0.950749	0.974937
	R	0.846154	0.676343	0.892598	0.686502	0.49492	0.587808	0.644412	0.564586
	F	0.906687	0.756493	0.869258	0.809239	0.660213	0.735695	0.768166	0.715074
6	OK	773	759	808	770	679	727	784	731
	FP	4	0	9	0	1	0	5	2
	FN	87	101	52	90	181	133	76	129
	D	0	0	0	0	0	0	1	1
	M	0	0	0	0	0	0	1	1
	MR	0	0	0	0	0	0	1.31579	0.775194
	DR	0	0	0	0	0	0	20	50
	MD	0.015016	-0.0087159	-0.0049965	-0.0059076	-0.00659667	-0.0133873	-0.0121436	-0.00669378
	P	0.994852	1	0.988984	1	0.998529	1	0.993663	0.997271
	R	0.898837	0.882558	0.939535	0.895349	0.789535	0.845349	0.911628	0.85
	F	0.944411	0.937616	0.963626	0.944785	0.881818	0.916194	0.950879	0.917765
7	OK	811	797	833	819	784	791	832	795
	FP	3	1	86	1	1	1	41	11
	FN	57	71	35	49	84	77	36	73
	D	0	0	0	0	0	1	0	0
	M	0	0	0	0	0	1	0	0
	MR	0	0	0	0	0	1.2987	0	0
	DR	0	0	0	0	0	100	0	0
	MD	0.0158608	-0.0069206	-0.0022457	-0.0035563	-0.00511601	-0.0112332	-0.00965005	-0.00450854
	P	0.996315	0.998747	0.90642	0.99878	0.998726	0.998737	0.953036	0.986352
	R	0.934332	0.918203	0.959677	0.943548	0.903226	0.91129	0.958525	0.915899
	F	0.964328	0.956783	0.932289	0.970379	0.948578	0.953012	0.955773	0.949821

Nº	P	UA	QMAW	QMCD	ACDD	AHQ	AKL	APD	ASD
8	OK	602	590	650	611	513	570	624	572
	FP	2	1	23	2	1	1	3	2
	FN	90	102	42	81	179	122	68	120
	D	0	0	0	1	1	1	1	1
	M	0	0	0	1	1	1	1	1
	MR	0	0	0	1.23457	0.558659	0.819672	1.47059	0.833333
	DR	0	0	0	50	100	100	33.3333	50
	MD	0.0173234	-0.0082776	-0.0043677	-0.0049846	-0.00637224	-0.0122655	-0.0106718	-0.00541229
	P	0.996689	0.998308	0.965825	0.996737	0.998055	0.998249	0.995215	0.996516
	R	0.869942	0.852601	0.939306	0.882948	0.741329	0.823699	0.901734	0.82659
	F	0.929012	0.919719	0.952381	0.936398	0.850746	0.902613	0.946171	0.903633
9	OK	856	931	956	693	606	632	808	586
	FP	6	0	18	0	0	0	0	0
	FN	214	139	114	377	464	438	262	484
	D	0	0	1	0	0	0	0	0
	M	0	0	1	0	0	0	0	0
	MR	0	0	0.877193	0	0	0	0	0
	DR	0	0	5.55556	0	0	0	0	0
	MD	0.0220496	-0.0089559	-0.0034661	-0.0026141	-0.0051246	-0.00942686	-0.0076768	-0.0037776
	P	0.993039	1	0.98152	1	1	1	1	1
	R	0.8	0.870093	0.893458	0.647664	0.566355	0.590654	0.75514	0.547664
	F	0.886128	0.930535	0.935421	0.78616	0.72315	0.742656	0.86049	0.707729
10	OK	600	591	656	591	518	536	605	530
	FP	5	0	17	1	0	0	19	2
	FN	110	119	54	119	192	174	105	180
	D	0	0	2	1	0	0	0	0
	M	0	0	2	1	0	0	0	0
	MR	0	0	3.7037	0.840336	0	0	0	0
	DR	0	0	11.7647	100	0	0	0	0
	MD	0.0184832	-0.0078105	-0.0036921	-0.0035451	-0.00494931	-0.0105561	-0.00915047	-0.00470025
	P	0.991736	1	0.97474	0.998311	1	1	0.969551	0.996241
	R	0.84507	0.832394	0.923944	0.832394	0.729577	0.75493	0.852113	0.746479
	F	0.912548	0.908532	0.948662	0.907834	0.843648	0.860353	0.907046	0.853462
11	OK	550	586	659	539	511	531	591	487
	FP	47	161	203	43	0	0	51	29
	FN	141	105	32	152	180	160	100	204
	D	0	0	7	20	0	0	3	16
	M	0	0	7	20	0	0	3	16
	MR	0	0	21.875	13.1579	0	0	3	7.84314
	DR	0	0	3.44828	46.5116	0	0	5.88235	55.1724
	MD	0.0252453	0.00669077	0.00422746	0.00654799	0.0101701	0.00128667	0.00094206	0.00542709
	P	0.921273	0.784471	0.764501	0.926117	1	1	0.920561	0.943798
	R	0.795948	0.848046	0.95369	0.780029	0.739508	0.768452	0.855282	0.704776
	F	0.854037	0.815021	0.84868	0.846819	0.85025	0.869067	0.886722	0.806959
12	OK	579	657	662	593	485	512	668	538
	FP	37	251	207	55	0	2	77	41
	FN	240	162	157	226	334	307	151	281
	D	0	0	3	26	0	1	7	21
	M	0	0	3	26	0	1	7	21
	MR	0	0	1.91083	11.5044	0	0.325733	4.63576	7.47331
	DR	0	0	1.44928	47.2727	0	50	9.09091	51.2195
	MD	0.0273481	0.00741835	0.00962537	0.00904399	0.0106961	0.00226111	0.00380457	0.00803303
	P	0.939935	0.723568	0.761795	0.915123	1	0.996109	0.896644	0.929188
	R	0.70696	0.802198	0.808303	0.724054	0.592186	0.625153	0.815629	0.656899
	F	0.806969	0.760857	0.78436	0.808453	0.743865	0.768192	0.85422	0.769671
13	OK	461	531	567	471	435	441	554	425
	FP	62	264	301	30	0	0	42	23
	FN	130	60	24	120	156	150	37	166
	D	0	0	4	18	0	0	3	16
	M	0	0	4	18	0	0	3	16
	MR	0	0	16.6667	15	0	0	8.10811	9.63855
	DR	0	0	1.3289	60	0	0	7.14286	69.5652
	MD	0.0283894	0.00797938	0.010005	0.00859877	0.00997123	0.00229231	0.0035799	0.00709687
	P	0.881453	0.667925	0.653226	0.94012	1	1	0.92953	0.948661
	R	0.780034	0.898477	0.959391	0.796954	0.736041	0.746193	0.937394	0.71912
	F	0.827648	0.766234	0.777245	0.862637	0.847953	0.854651	0.933446	0.818094
14	OK	533	575	589	513	539	546	587	483
	FP	42	355	423	106	0	0	132	91
	FN	63	21	7	83	57	50	9	113
	D	0	0	10	23	0	0	5	17
	M	0	0	10	23	0	0	5	17
	MR	0	0	142.857	27.7108	0	0	55.5556	15.0442
	DR	0	0	2.36407	21.6981	0	0	3.78788	18.6813
	MD	0.0257345	0.00807489	0.00928323	0.00745599	0.00985965	0.00237859	0.0011278	0.00511899
	P	0.926957	0.61828	0.582016	0.828756	1	1	0.816412	0.841463
	R	0.894295	0.964765	0.988255	0.860738	0.904362	0.916107	0.984899	0.810403
	F	0.910333	0.753604	0.732587	0.844444	0.94978	0.956217	0.892776	0.825641

Nº	P	UA	QMAW	QMCD	ACDD	AHQ	AKL	APD	ASD
15	OK	456	559	565	516	442	441	559	461
	FP	52	258	363	14	0	0	31	11
	FN	116	13	7	56	130	131	13	111
	D	0	0	5	4	0	0	2	4
	M	0	0	5	4	0	0	2	4
	MR	0	0	71.4286	7.14286	0	0	15.3846	3.6036
	DR	0	0	1.37741	28.5714	0	0	6.45161	36.3636
	MD	0.0277738	0.00916181	0.00970534	0.00966588	0.00983424	0.00318401	0.00388047	0.00837891
	P	0.897638	0.684211	0.608836	0.973585	1	1	0.947458	0.976695
	R	0.797203	0.977273	0.987762	0.902098	0.772727	0.770979	0.977273	0.805944
	F	0.844444	0.804896	0.753333	0.936479	0.871795	0.870681	0.962134	0.883142

7. Sistema de Propagación de Interacciones.

Un sistema de propagación de interacciones es aquel que a su salida permite a un usuario interactuar con los resultados, aprovechando la información que este le facilita para producir una nueva salida adaptando sus parámetros a la nueva información. De esta manera el sistema aprovecha la mejor información de la que se dispone, en este caso un oyente real, para obtener la detección perfecta mediante el menor numero de interacciones posible.

7.1. Introducción.

Como se ha podido observar a lo largo de este trabajo, el esquema general de detección automática de onsets consiste en la construcción de una función de detección a la cual se le aplica un detector de picos, de cuales serán onsets los que superen cierto umbral. Con las limitaciones psicocacústicas conocidas, la detección perfecta sería aquella en la que los onsets indicados por un oyente experto fueran exactamente los picos de la ODF que han superado el umbral establecido. De la experimentación se concluye que cada sistema tiene un umbral "base" diferente, según como este construida su ODF, y que buscar un umbral adecuado a un sistema de detección concreto no es fácil y requiere un método de ensayo y error hasta dar con la mejor detección posible. A esto hay que sumarle, que aún tratándose del mismo sistema de detección, el umbral adecuado también será diferente según la naturaleza de la señal en estudio e incluso distintos tramos dentro de una misma señal (cambios de instrumentación, cambios de rango de dinámica,...). En la literatura, como ya se ha explicado, se han intentado diversas técnicas que palien el efecto del establecimiento de un umbral para discernir los picos de la ODF que son onsets. El método ideal sería aquel en el que la ODF fuera perfecta y todos sus picos fueran onsets. Esto aún no se ha conseguido, por lo que se han implementado técnicas como la umbralización dinámica, la normalización adaptativa, etc, que en muchos casos mejoran la detección pero sin llegar a ser perfecta.

El sistema de propagación de interacciones presenta a un usuario experto la salida de un sistema de detección automática de onsets, y recoge el feedback que este le proporciona para recalcular su salida mediante la variación de sus parámetros en base a la nueva información. Mediante este recálculo la corrección para obtener la detección perfecta debería necesitar muchas menos interacciones con el usuario que una corrección manual.

Como ejemplo, el parámetro a modificar en el sistema de detección de la UA es el threshold del detector de picos. A priori, como se ha visto en la sección de resultados de la evaluación del sistema, un threshold "base" que da buenos resultados en la mayoría de casos es 0´18. Un usuario ajeno a un sistema no tiene por que conocer este dato. El sistema de propagación de interacciones solucionaría este problema en el momento en el que un

usuario detecte un falso positivo o un falso negativo en su salida, ya que establecerá un nuevo threshold en función de esta información.

En un sistema normal, el usuario informa el threshold y sistema de detección le dice donde están localizados los onsets. Por el contrario, haciendo uso de un sistema de propagación a la salida del sistema de detección, el usuario indicará donde hay un onsets en caso de falso negativo, o donde no lo hay en caso de falso positivo, y será el sistema el que establezca el nuevo threshold y recalcule su salida en base a este para eliminar errores similares. De esta forma, aún estableciendo un umbral no idóneo, en la primeras interacciones el sistema "afinará" el umbral necesario para una detección perfecta.

Lo que se pretende con este sistema de propagación de interacciones es minimizar el número de correcciones necesarias para obtener una detección ideal, frente a un sistema de detección que no use propagación.

7.2. Metodología.

En el caso que nos ocupa, para implementar un sistema de propagación automático se necesitan los siguientes elementos:

- Información de un usuario experto para corregir la salida del sistema de detección.
- Salida del sistema de detección.
- Picos de la función de la detección.
- Margen de error temporal para considerar una detección correcta.
- Umbral inicial.

La información que un usuario daría a la salida del sistema de detección ya estaría establecida en los onsets etiquetados de forma manual, puesto que son los onsets reales que ha señalado el oyente. Usaremos esta información para construir una propagación automática. Para ello, además de la información de onsets reales necesitamos los onsets etiquetados por el sistema y la función de detección construida con los mismos parámetros que la detección realizada. Esto es necesario para que la resolución temporal de la ODF corresponda con la resolución temporal de la detección. Esta función de detección hay que

procesarla a través de un detector de picos, puesto que solo los picos de la función de detección son candidatos a onsets, el resto de valores será 0.

Normalmente para evaluar un sistema de detección de onsets de forma automática se analiza de manera independiente la información para buscar falsos positivos y falsos negativos. Los falsos positivos se buscan en la salida del sistema comparando con aquellos onsets que no existen en el groundtruth. Por el contrario, los falsos negativos se buscan el el grountruth comparando con aquellos onsets que no existen en la salida del sistema. Este sistema no es útil para el estudio del sistema de propagación, puesto que el fin último del mismo es interactuar con el usuario, por lo que ambos casos (FP y FN) deben localizarse y corregirse de manera simultánea según el usuario avanza en la corrección de la salida del sistema. De este modo, se contará cada corrección del usuario como una interacción, aunque esta corregirá ese error y todos los similares, minimizando el número de correcciones a realizar por el usuario.

7.2.1. Algoritmo de propagación de corrección de errores para un sistema automático de detección de onsets.

Como se deduce de la metodología para implementar el algoritmo se necesitan, entre otros, los siguientes elementos indispensables:

Vectores:

- OnsetsManuales --> contiene instantes de tiempo de onsets en segundos etiquetados por un oyente humano.
- OnsetsDetectados --> contiene instantes de tiempo de onsets en segundos detectados por el sistema.
- FunciondeDetección --> contiene todos los instantes de tiempo en segundos según la resolución del sistema y con todos los valores = 0 excepto los picos de la función de detección.

Variables:

- Margen de error --> margen de error para considerar un onsets como correcto. En este caso se ha usado 50ms --> 0,05s.
- Número de interacciones --> contador para el numero de interacciones que corregirá el usuario.

- Threshold --> Umbral del sistema de detección.

Desarrollo:

- Se inicia una iteración que recorre el vector de onsets manuales.
- Se localiza el primer elemento.
- Se comprueba si entre 0 s y el tiempo del primer onsets maual - margen de error, existe algún onset en el vector de onsets detectados.
- Si existe/n, es/son Falso/s Positivo/s.
- Si es un Falso Positivo --> procedimiento para corrección de Falsos Positivos.
- Se comprueba si en el tiempo indicado por el primer elemento del vector de onsets manuales +- el margen de error establecido existe algún elemento en el vector de onsets detectados.
- Si existe, es un onsets correcto.
- Si no existe, es un Falso negativo.
- Si es un Falso Negativo --> procedimiento de corrección de Falsos Negativos.
- Se continua la iteración hacia el siguiente elemento del vector de onsets manuales.
- Se repite el preceso hasta el final del vector de onsets manuales, con la excepción de que a partir de esta iteración los Falsos Positivos se buscan entre el tiempo del elemento de onsets manual actual - margen de error y el tiempo del elemento anterior de onsets manual + margen de error.

- Procedimiento de corrección de Falsos Positivos:
  - Puesto que el vector ODF y el vector de onsets detectados tienen la misma resolución temporal, se busca el elemento del vector ODF que corresponde al tiempo exacto del Falso Positivo.
  - Se comprueba la energía del pico de este elemento en el vector ODF y se establece este valor como nuevo umbral.
  - A partir de este elemento del vector ODF se recorre el mismo comprobando las energías de cada pico.
  - Si en la ODF se encuentra, para un valor de tiempo, un valor de energía menor o igual al nuevo umbral, se busca el tiempo exacto correspondiente en el vector de onsets detectados y se elimina este elemento redimensionando el vector.

- Procedimiento de corrección de Falsos Negativos:
  - Se localiza en el vector ODF el tiempo del pico que se encuentre entre el tiempo del elemento actual de onsets manuales +- margen de error. Notar que como mucho solo puede ser un pico uno de cada tres elementos de la ODF, por lo que en principio y según los parámetros de resolución temporal del algoritmo de detección solo habrá un pico que cumpla esta condición.
  - Se comprueba el valor de energía de dicho elemento del vector ODF y se establece como nuevo threshold.
  - A partir de este elemento se recorre el vector ODF en busca de los picos que sean mayores o iguales que el nuevo umbral.
  - Por cada elemento que cumpla esta condición, se inserta en la posición del vector de onsets detectados que le corresponda el valor de tiempo exacto localizado en el vector ODF.

7.2.3 Código.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

using namespace std;

const double kDEVIATION=0.050; // Margen de error para onsets detectados

typedef struct {
  double time;
  double energy;
} ODF;

bool falsesPositives(double a, double b, double c)
{
	return (b>=a+kDEVIATION && b<=c-kDEVIATION);
}

bool onsetOK(double a, double b)
{
	return (b>=a-kDEVIATION && b<=a+kDEVIATION);
}

int searchPosition(double time, vector<double> vd, vector<ODF> vdf, bool odf)
{
	bool encontrado = false;
	int aux = 0;

	if(!odf)
	{
		for(int i=0; i<vd.size() && !encontrado; i++)
		{
			if(vd[i] >= time-kDEVIATION)
```

```
                                {
                                        aux = i;
                                        encontrado = true;
                                }
                        }
                }
                else
                {
                        for(int i=0; i<vdf.size() && !encontrado; i++)
                        {
                                if(vdf[i].time >= time-kDEVIATION && vdf[i].time <= time+kDEVIATION)
                                {
                                        aux = i;
                                        encontrado = true;
                                }
                        }
                }

                return aux;
}

bool exist(double time, vector<double> vdetected)
{
                bool exist = false;
                for(int i=0; i<vdetected.size() && !exist; i++)
                {
                        if(onsetOK(time,vdetected[i]))
                                exist = true;
                }
                return exist;
}

void insertTimeInDetected(double time, vector<double>& vdetected)
{
                vector<double>::iterator actual=vdetected.begin();
                bool notInserted = true;

                for(actual; actual<vdetected.end() && notInserted; actual++)
                {
                        if(time <= *actual)
                        {
                                actual = vdetected.insert(actual,time);
                                notInserted = false;
                        }
                }
}

void insertTimesInDetected(double energy, vector<ODF> vdfunction, vector<double>& vdetected)
{
                int aux = 0;

                for(int i=0; i<vdfunction.size(); i++)
                {
                        if(vdfunction[i].energy == energy)
                                aux = i;
                }
                for(int i=aux; i<vdfunction.size(); i++)
                {
                        if(energy <= vdfunction[i].energy)
                        {
                                if(!exist(vdfunction[i].time,vdetected))
                                        insertTimeInDetected(vdfunction[i].time,vdetected);

                                cout << "INSERTANDO... " << "VALOR INSERTADO: " << vdfunction[i].time <<
endl;
                        }
                }
}
```

```
void searchPositionInDetected(double vdfunctiontime, vector<double> vdetected, int& aux2, bool& existe_en_detected)
{
	bool notfound = true;
	for(int i=0; i<vdetected.size() && notfound; i++)
	{
		if(vdetected[i] == vdfunctiontime)
		{
			aux2 = i;
			notfound = false;
		}
	}
	if(!notfound)
		existe_en_detected = true;
}

void deleteTimeInDetected(int j, vector<double>& vdetected)
{
	vdetected.erase(vdetected.begin()+j);
}

void deleteTimesInDetected(double energy, int position, vector<ODF> vdfunction, vector<double>& vdetected)
{
	int aux = 0;
	int aux2 = 0;
	bool existe_en_detected = false;

	for(int i=position; i<vdfunction.size(); i++)
	{
		if(energy >= vdfunction[i].energy)
		{
			searchPositionInDetected(vdfunction[i].time,vdetected,aux2,existe_en_detected);
			if(existe_en_detected)
			{
				cout << "BORRANDO... " << "VALOR BORRADO: " << vdetected[aux2] << endl;
				deleteTimeInDetected(aux2,vdetected);
				existe_en_detected = false;
			}
		}
	}
}

void searchExactPosition(double time, vector<ODF> vdfunction, double& threshold, int& position)
{
	bool breakLoop = false;

	for(int i = 0; i<vdfunction.size() && !breakLoop; i++)
	{
		if(vdfunction[i].time == time)
		{
			threshold = vdfunction[i].energy;
			position = i;
			cout << "threshold nuevo: " << vdfunction[i].energy << endl;
			cout << "posicion en odf: " << position << endl;
			breakLoop = true;
		}
	}
}

void boomInDetected(double time, vector<double>& vdetected)
{
	int lastpositives = 0;
	for(int i = 0; i<vdetected.size(); i++)
	{
		if(vdetected[i]>time+kDEVIATION)
		{
			cout << "Borrando tiempo : " << vdetected[i] << endl;
			lastpositives++;
```

```
                }
        }
        vdetected.erase(vdetected.end()-lastpositives,vdetected.end());
}

void propagation(vector<double> voriginal, vector<double>& vdetected, vector<ODF> vdfunction, int& numfpinteracciones, int& numfninteracciones, double originalthreshold)
{
        bool buclej = true;
        bool odf = false;
        int j = 0;
        int k = 0;

        int position = 0;

        double threshold = originalthreshold;

        for(int i=0; i<voriginal.size(); i++)
        {
                j = searchPosition(voriginal[i],vdetected,vdfunction,odf); // iterador FN

                // ZONA DE FALSOS POSITIVOS
                if(i!=0)
                {
                        for(int z = 0; z<vdetected.size(); z++)
                        {
                                if(falsesPositives(voriginal[i-1],vdetected[z],voriginal[i]))
                                {
                                        numfpinteracciones++;
                                        cout << "FALSO POSITIVO: " << vdetected[z] << endl;
                                        searchExactPosition(vdetected[z],vdfunction,threshold,position);
                                        deleteTimesInDetected(threshold,position,vdfunction,vdetected);
                                }
                        }
                }
                else
                {
                        for(int it=0; it<vdetected.size(); it++)
                        {
                                if(vdetected[it] < voriginal[i]-kDEVIATION)
                                {
                                        searchExactPosition(vdetected[it],vdfunction,threshold,position);
                                        if(threshold == 0)
                                                deleteTimeInDetected(position,vdetected);
                                        else

        deleteTimesInDetected(threshold,position,vdfunction,vdetected);
                                        it--;
                                }
                        }
                }

                // ZONA FALSOS NEGATIVOS
                if(onsetOK(voriginal[i],vdetected[j]))
                {
                }
                else // FALSO NEGATIVO
                {
                        numfninteracciones++;
                        odf = true;
                        k = searchPosition(voriginal[i],vdetected,vdfunction,odf);
                        odf = false;

                        if(vdfunction[k].energy == 0)
                        {
                                threshold = originalthreshold;
                                if(!exist(voriginal[i],vdetected))
                                {
                                        insertTimeInDetected(voriginal[i],vdetected);
```

```
                                        }
                                }
                                else
                                {
                                        cout << endl << "FALSO NEGATIVO: " <<voriginal[i] << endl;
                                        threshold = vdfunction[k].energy;
                                        cout << "threshold nuevo: " << threshold << endl;
                                        cout << "posicion en odf: " << k << endl;
                                        insertTimesInDetected(threshold,vdfunction,vdetected);
                                }
                        }

                        if(i+1==voriginal.size())
                        {
                                boomInDetected(voriginal[i],vdetected);
                        }
                }
}

int main(int argc, char *argv[]) {

        if (argc!=5) {
                cerr << "Sintax: " << argv[0] << " <original_onsets.txt> <detected_onsets.txt>
<onsets_detection_function.txt> <threshold>" << endl;
                exit(-1);
        }

        ifstream inoriginal(argv[1]);
        ifstream indetected(argv[2]);
        ifstream indfunction(argv[3]);
        ofstream  fo;
        vector<double> voriginal;
        vector<double> vdetected;
        vector<ODF> vdfunction;

        int numfninteracciones = 0;
        int numfpinteracciones = 0;

        if (inoriginal.is_open())
        {
           double line=0.0;
           while (!inoriginal.eof())
           {
                inoriginal >> line;
                        voriginal.push_back(line);
                }
                if (indetected.is_open())
                {
                while (!indetected.eof())
                {
                        indetected >> line;
                                vdetected.push_back(line);
                        }
             indetected.close();
                }
                else cerr << "Error: file " << argv[2] << " not found\n";
                if(indfunction.is_open())
                {
                        ODF element;
                        element.time = 0.0;
                        element.energy = 0.0;

                        while(!indfunction.eof())
                        {
                                indfunction >> element.time;
                                indfunction >> element.energy;
                                vdfunction.push_back(element);

                        }
```

```
                        indfunction.close();

                        cout << "Número de onsets groundtruth: " << voriginal.size() << endl;
                        cout << "Número de onsets detectados inicialmente: " << vdetected.size() << endl
<< endl;


        propagation(voriginal,vdetected,vdfunction,numfpinteracciones,numfninteracciones,atof(argv[4]));

                        cout << endl << "Número de onsets detectados final: " << vdetected.size() << endl;

                        cout << endl << "número interacciones FP: " << numfpinteracciones << endl;
                        cout << "número interacciones FN: " << numfninteracciones << endl;

                        fo.open("Onsets Propagación.txt",ios::out);
                        if(fo.is_open())
                        {
                                for(int i=0; i<vdetected.size(); i++)
                                {
                                        fo << vdetected[i] << "\n";
                                }
                                fo.close();
                        }
                }
                else cerr << "Error: file " << argv[3] << " not found\n";
                inoriginal.close();
        }
        else cerr << "Error: file " << argv[1] << " not found\n";
}
```

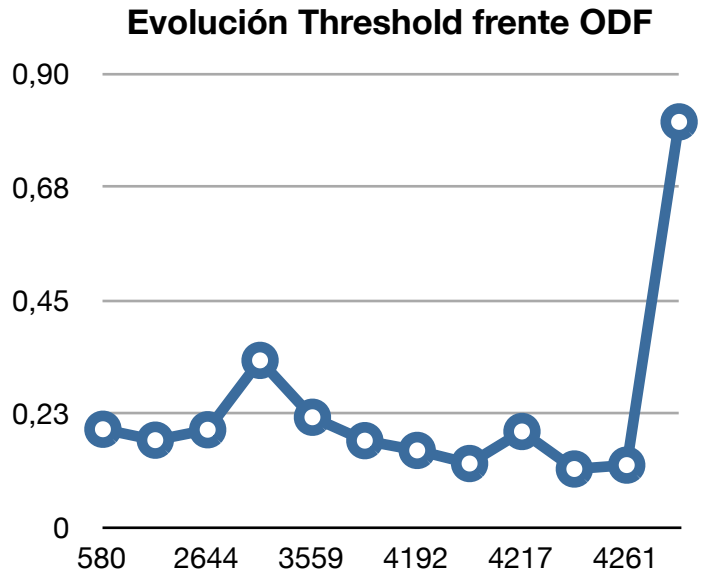
7.3. Resultados.

Para evaluar el funcionamiento del algoritmo de propagación de interacciones se realiza un banco de pruebas aplicando el mismo a las 15 señales que componen la base de datos que se ha elaborado en este trabajo. Los archivos de etiquetas de onsets para establecer el groundtruth son los elaborados en el punto 5.2 de la presente memoria, los archivos de onsets detectados se obtienen mediante la detección de señales de audio en el plug-in Vamp del detector de onsets de la Universidad de Alicante en Sonic Visualizer, y por último los archivos de funciones de detección se obtienen también del citado plug-in y han sido sometidos a una detección de picos.

En la siguiente tabla, N es el numero de la señal en estudio, GT es el número de onsets etiquetados manualmente, OD es el número de onsets a la salida del sistema de detección, OP es el número de onsets después de la propagación de la corrección, FDP es el número de falsos positivos del sistema de detección, FND es el número de falsos negativos a la salida del sistema de detección, FPP es el número de falsos positivos corregidos por el usuario en propagación, FNP es el número de falsos negativos corregidos por el usuario en propagación, TFD es la suma de falsos positivos y falsos negativos en la detección, TFP es la suma de falsos positivos y falsos negativos en propagación, ID es el número total de interacciones del usuario para corregir la detección de forma manual, IP es el número total de interacciones para corregir la detección mediante propagación.

N°	GT	OD	OP	FPD	FND	FPP	FNP	TFD	TFP	ID	IP
1	858	756	821	37	139	23	209	176	232	278	269
2	845	799	837	19	65	16	105	84	121	130	129
3	702	622	657	38	118	23	148	156	171	236	216
4	994	831	944	45	208	32	269	253	301	416	351
5	690	598	667	14	106	20	168	120	188	212	211
6	861	778	822	4	87	5	60	91	65	174	104
7	869	815	843	3	57	2	87	60	89	114	115
8	693	605	656	2	90	3	54	92	57	180	94
9	1071	863	1055	6	214	4	273	220	277	428	293
10	711	606	678	5	110	5	127	115	132	220	165
11	692	598	685	47	141	27	235	188	262	282	269
12	820	617	705	37	240	16	279	277	295	480	410
13	592	524	587	62	130	14	377	192	391	260	396
14	597	576	597	42	63	12	235	105	247	126	247
15	573	509	569	52	116	8	397	168	405	232	409

La siguiente gráfica muestra la evolución del threshold a lo largo de los frames de la ODF para la señal número 6:



7.4. Conclusiones.

El primer punto a destacar una vez analizados los resultados arrojados en el banco de pruebas realizado, es la no consecución de la corrección perfecta en el sistema de propagación. Notar que para una corrección perfecta el número de onsets arrojados por el sistema de propagación debería coincidir con el número de onsets reales, y no es el caso excepto en la señal Nº 14. Esto puede ser debido a que la resolución temporal de la ODF y el margen de error establecido para considerar un onsets como correcto, estén dando lugar a onsets doblados o mezclados. Por norma general el sistema de propagación rebaja el número de interacciones con el usuario en la corrección, aunque no en la proporción en que se espera en la teoría. Incluso en algunas señales el número de interacciones con propagación es mayor que en una corrección manual. Para solucionar esto y determinar el posible fallo del algoritmo, sería interesante analizar de nuevo las etiquetas de onsets a la salida del sistema de propagación frente al groundtruth establecido. En la siguiente tabla se muestran los resultados de la evaluación de la salida del sistema de propagación frente a los onsets reales:

Señal 1 OK= 818 FP= 3 FN= 39 Doubled= 1 Merged= 1 MergRate= 2.564 DouRate= 33.333 MDtion= 0.01441 Prec= 0.996346 Rec= 0.954492 Fmeasure= 0.974	Señal 2 OK= 835 FP= 2 FN= 9 Doubled= 0 Merged= 0 MergedRate= 0 DoubledRate= 0 MDtion= -0.0004 Prec= 0.997611 Rec= 0.989336 Fmeasure= 0.993	Señal 3 OK= 653 FP= 4 FN= 48 Doubled= 2 Merged= 2 MergedRate= 4.1666 DoubledRate= 50 MeanDetion= 0.0146 Prec= 0.993912 Rec= 0.931526 Fmeasure= 0.961708	Señal 4 OK= 941 FP= 3 FN= 52 Doubled= 3 Merged= 3 MergedRate= 5.76923 DoubledRate= 100 MeanDeviation= 0.0135 Prec= 0.996822 Rec= 0.947633 Fmeasure= 0.971606	Señal 5 OK= 666 FP= 1 FN= 23 Doubled= 0 Merged= 0 MergedRate= 0 DoubledRate= 0 MeanDeviation= 0.0048 Prec= 0.998501 Rec= 0.966618 Fmeasure= 0.982301
Señal 6 OK= 822 FP= 0 FN= 38 Doubled= 0 Merged= 0 MergedRate= 0 DoubledRate= 0 MDtion= 0.01417 Prec= 1 Rec= 0.955814 Fmeasure= 0.977	Señal 7 OK= 842 FP= 1 FN= 26 Doubled= 1 Merged= 1 MergRate= 3.846 DoubledRate= 100 MeDtion= 0.0142 Prec= 0.998814 Rec= 0.970046 Fmeasure= 0.984	Señal 8 OK= 655 FP= 1 FN= 37 Doubled= 0 Merged= 0 MergedRate= 0 DoubledRate= 0 MeanDetion= 0.0162 Prec= 0.998476 Rec= 0.946532 Fmeasure= 0.97181	Señal 9 OK= 1053 FP= 2 FN= 17 Doubled= 2 Merged= 2 MergedRate= 11.7647 DoubledRate= 100 MeanDeviation= 0.0163 Prec= 0.998104 Rec= 0.984112 Fmeasure= 0.991059	Señal 10 OK= 677 FP= 1 FN= 33 Doubled= 1 Merged= 1 MergedRate= 3.0303 DoubledRate= 100 MeanDeviation= 0.0151 Prec= 0.998525 Rec= 0.953521 Fmeasure= 0.975504
Señal 11 OK= 684 FP= 1 FN= 7 Doubled= 1 Merged= 1 MergRate= 14.28 DoubleRate= 100 MDtion= 0.01652 Prec= 0.99854 Rec= 0.98987 Fmesure= 0.9941	Señal 12 OK= 705 FP= 0 FN= 114 Doubled= 0 Merged= 0 MergedRate= 0 DoubledRate= 0 MDtion= 0.01769 Prec= 1 Rec= 0.860806 Fmesure= 0.9251	Señal 13 OK= 585 FP= 2 FN= 6 Doubled= 2 Merged= 2 MergedRate= 33.333 DoubledRate= 100 MeanDetion= 0.0104 Prec= 0.996593 Rec= 0.989848 Fmeasure= 0.993209	Señal 14 OK= 596 FP= 1 FN= 0 Doubled= 1 Merged= 1 MergedRate= 0 DoubledRate= 100 MeanDeviation= 0.0151 Prec= 0.998325 Rec= 1 Fmeasure= 0.999162	Señal 15 OK= 568 FP= 1 FN= 4 Doubled= 1 Merged= 1 MergedRate= 25 DoubledRate= 100 MeanDeviation= 0.0087 Prec= 0.998243 Rec= 0.993007 Fmeasure= 0.995618

8. Conclusiones y trabajo futuro.

La detección automática de onsets es una poderosa herramienta en el campo de la informática musical y tiene, como se explicado a lo largo de este trabajo un gran número de campos de aplicación, siendo el sistema sobre el que se construyen muchas otras aplicaciones. A día de hoy la detección automática de onsets sigue siendo un problema abierto. Existen multitud de enfoques al problema desde hace mas de 20 años, pero a día de hoy aunque se ha mejorado bastante en porcentajes de detección correcta gracias a la potencia de los ordenadores en la actualidad, aún no existe un detector perfecto. La naturaleza y características de las señales musicales es muy amplia y diferente y no todos los detectores funcionan bien en todos los casos. Hay una tendencia a implementar sistemas que incluyan varias construcciones de ODF para que sea el usuario quien determine cual ejecutar en cada señal. Este es el otro gran handicap de la detección automática de onsets: su valoración es muy subjetiva, ya que depende de un oyente humano. Además, cualquier oyente no es valido para valorar un sistema o establecer un groundtruth, ya que se necesitan ciertos conocimientos, aptitudes y entrenamiento para hacerlo de forma correcta, por lo que se limita a usuarios expertos. Por otro lado, de los resultados obtenidos en el banco de pruebas del sistema de propagación de errores no se pueden sacar conclusiones determinantes, ya que como se ha explicado el número de erroes en la corrección no coincide con el número de onsets reales. Como trabajo futuro, se debe investigar la causa de esta incongruencia y determinar si es debido a una incorrecta implementación del algoritmo, a los parámetros establecidos, o si se debe al propio concepto de propagación de errores. También como trabajo futuro es conveniente evaluar el efecto de la propagación en detecciones realizadas con umbrales diferentes, que permita observar cuantas interacciones son necesarias para establecer un umbral válido. Otra vía de trabajo es aplicar el sistema de propagación a algoritmos de detección que tengan un parámetro similar a un threshold para poder variar en función de la información proporcionada por el usuario, y a los cuales tengamos acceso para obtener una detección y su función de detección.

9. Bibliografía y fuentes de información.

1. Pérez-García Tomás, Iñesta José M., Pedro Ponce de León J., PertuSA Antonio "A multimodal Music Transcription Prototype, Frist steps in an interactive prototype development" Proc. Of International Conference on Multimodal Interaction, ICMI 2011, ISBN: 978-1-4503-0641-6, pp. 315—318, Alicante, Spain (2011).
2. Pertusa Antonio, Klapuri Anssi, Iñesta josé M. "Recognition of Note Onsets in Digital Music Using Semitone Bands" Progress in Pattern Recognition, Image Analysis and Aplications: 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005. Havana, Cuba. LNCS, Volume 3773/2005, pages 869-879. Eds. Alberto Sanfeliu, Manuel Lazo.
3. Pertusa Antonio, Iñesta José M. "Efficient methods for join estimation of multiple fundamental frequencies in music signal" EURASIP Journal on Advances in Signal Progressing, vol. 2012, pp. 27 (2012).
4. Argenti Fabrizio, Nesi Paolo, Pantaleo Gianni "Automatic Music Transcription: From Monophonic to Polyphonic" Musical Robots and Interactive Multimodal Systems Springer 2011.
5. J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005.
6. Roads Curtis " The Computer Music Tutorial" The Mitt Press 1996.
7. Proakis John G., Manolakis Dimitri G. Tratamiento digital de señales Pearson Prentice Hall 2007.
8. Zölzer Udo "Digital Audio Signal Processing".
9. Prototipo http://miprcv.iti.upv.es/index.php?option=com_content&task=view&id=230&Itemid=220

10. Klapuri, A. "Sound Onset Detection by Applying Psychoacoustic Knowledge", *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, March 15-19, 1999, Phoenix, USA, pp. 3089-3092
11. Goto, M. and Muraoka, Y. "Beat tracking based on multiple-agent architecture —A real-time beat tracking system for audio signals —" in *Proc. of the Second Int. Conf. on Multi-Agent Systems*, pp.103–110, December 1996.
12. Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M. and Sandler, M.B. "A tutorial on onset detection in music signals", in *IEEE Transactions on Speech and Audio Processing"*, vol. 13, issue 5, pp. 1035 – 1047, Sept. 2005.
13. Scheirer, E.D. "Tempo and beat analysis of acoustic musical signals" *J. Acoust. Soc. Am.*, vol. 103, no.1, pp. 588-601, Jan 1998
14. Duxbury, C., Sandler, M. and Davies, M. "A hybrid approach to musical note onset detection" in *Proc. Digital Audio Effects Conference (DAFX)*, 2002.
15. Goto, M. and Muraoka, Y. "A Real-Time Beat Tracking System for Audio Signals" *Proc. of the 1995 Int. Computer Music Conference*, pp. 171–174, Sep 1995 Recognition of Note Onsets in Digital Music Using Semitone Bands 879
16. Bilmes, J. "Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning and Reproducing Expressive Timing in Percusive Rhythm". MSc Thesis, MIT, 1993.

17. Goto, M. ”RWC music database”, published at http://staff.aist.go.jp/m.goto/RWC-MDB/
18. Moore, B.C.J., ”An introduction to the Psychology of Hearing”, Academic Press, fifth edition, 1997.
19. Young, S., Kershaw, D, Odell, J., Ollason, D., Valtchev, V. and Woodland, P. ”The HTK book (for HTK version 3.1)” *Cambridge University*, 2000.
20. Rodet, X., Escribe, J. and Durignon, S. ”Improving score to audio alignment: Percussion alignment and Precise Onset Estimation” *Proc. of the 2004 Int. Computer Music Conference*, pp. 450–453, Nov. 2004.
21. Lerch, A., Klich, I. ”On the Evaluation of Automatic Onset Tracking Systems”, *White Paper, Berlin, Germany, April 2005*.
22. J.P. Bello, *.Towards the Automated Analysis of SImple Polyphonic Music: A Knowledge-based Approach..*, PhD Diss., Queen Mary, University of London, 2003.
23. C. Duxbury, M. Davies, and M. Sandler, .Improved Time-Scaling of Musical Audio Using Phase Locking at Transients,. in *Proc. AES 112th Convention*, 2002.
24. P. Masri, *Computer Modeling of Sound for Transformation and Synthesis of Musical Signals*, PhD Thesis, University of Bristol, 1996.
25. Xavier Rodet and Florent Jaillet, .Detection and modeling of fast attack transients,. in *Proceedings of the International Computer Music Conference*, 2001.
26. Klapuri, .Sound Onset Detection by Applying Psychoacoustic Knowledge,. In *Proc. IEEE Conf. Acoustics, Speech and Signal Proceesing (ICASSP,'99)*, 1999.
27. Juan P. Bello and Mark Sandler, .Phase-based note onset detection for music signals,. in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-03*, 2003.
28. C. Duxbury, J.P. Bello, M. Davies, and M. Sandler, .A combined phase and amplitude based approach to onset detection for audio segmentation,. in *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS-03), London, UK.*, 2003.
29. Kauppinen, .Methods for detecting impulsive noise in speech and audio aignals,. in *Proc. DSP2002*.
30. N. G. Kingsbury, .The dual-tree complex wavelet transform: a new technique for shift invariance and directional _lters.,. in *Proc. IEEE Digital Signal Processing Workshop, DSP 98, Bryce Canyon UT*, 1998.
31. B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 5th ed. New York: Academic, 1997.
32. C. Duxbury, M. Sandler, and M. Davies, “A hybrid approach to musical note onset detection,” in *Proc. Digital Audio Effects Conf. (DAFX,’02)*, Hamburg, Germany, 2002, pp. 33–38.
33. J. Princen and A. Bradley, “Analysis/synthesis filter bank design based on time domain aliasing cancellation,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 5, pp. 1153–1161, Oct. 1986.
34. W. Schloss, “On the Automatic Transcription of Percussive Music—From Acoustic Signal to High-Level Analysis,” Ph.D. dissertation, Tech. Rep. STAN-M-27, Dept. Hearing and Speech, Stanford Univ., Stanford, CA, 1985.

35. Moore, B. Glasberg, and T. Bear, “A model for the prediction of thresholds, loudness and partial loudness,” *J. Audio Eng. Soc.*, vol. 45, no. 4, pp. 224–239, 1997.
36. M. Dolson, “The phase vocoder: a tutorial,” *Comput. Music J.*, vol. 10, no. 4, 1986.
37. J. P. Bello and M. Sandler, “Phase-based note onset detection for music signals,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-03)*, Hong Kong, 2003, pp. 49–52.
38. N. Collins. Using a pitch detector for onset detection. In Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 100–106, 2005.
39. S. Dixon. Onset detection revisited. In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 133–137, Montreal, Quebec, Canada, Sept. 18–20, 2006.
40. C. Duxbury, M. Sandler, and M. Davies. A hybrid approach to musical note onset detection. In Proceedings of the DAFx Conference, Hamburg, Germany, pages 33–38, 2002.
41. E. Kapanci and A. Pfeffer. A hierarchical approach to onset detection. In Proc. International Computer Music Conference (ICMC’04), pages 438–441, 2004.
42. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), volume 6, pages 3089–3092, 1999.
43. Lacoste and D. Eck. A supervised classification algorithm for note onset detection. EURASIP Journal on Applied Signal Processing, August 2007.
44. P. Leveau, L. Daudet, and G. Richard. Methodology and tools for the evaluation of automatic onset detection algorithms in music. In Proceedings of 5th International Symposium on Music Information Retrieval, pages 72–75, Barcelona, Spain, 2004.