

# Práctica Evaluable: Desarrollo de un sensor de lluvia y granizo

Asael Fernández Barcía  
*Audio Digital*  
Ingeniería de imagen y sonido en telecomunicación  
Cáceres, Cáceres  
afernandwg@alumnos.unex.es

Gerardo Pérez González  
*Audio Digital*  
Ingeniería de imagen y sonido en telecomunicación  
Cáceres, Cáceres  
geperezg@alumnos.unex.es

**Abstract**—El documento trata sobre el proceso de investigación y desarrollo de un algoritmo de lluvia y granizo a través del software matemático MATLAB, escogiendo una serie de parámetros considerados más apropiados. Al especificarse su uso en un entorno forestal, se ha enfocado el proyecto al mundo agrícola.

**Index Terms**—FFT, Densidad espectral de potencia, Short-Time energy, Centroide espectral, Flux espectral, Roll-off espectral

## I. INTRODUCCIÓN

Las precipitaciones son consideradas como el parámetro del clima agrícola más importante, y por otra parte, el más difícil de medir de manera precisa. Cada año es necesario tomar ciertas decisiones a la hora de organizar los cultivos que conllevan el movimiento de millones de euros, y de lo exhaustiva que sea la planificación puede depender que estos se pierdan o no [7].

Debido a la dificultad de medida de las precipitaciones se ha buscado desarrollar diversos sistemas que permitan disminuir el efecto negativo que puedan producir. El auge de los sistemas automatizados para su uso en la vida cotidiana se aprovecha como elemento para disminuir costes, ya que sistemas como las redes antigranizo suponen costes de alrededor de los 60.000 euros anuales solo en los referidos a las operaciones de manipulación (desplegar y plegar según época del año) [6].

La finalidad del sistema será distinguir entre el granizo y la lluvia, de tal manera que se permitirá regar de manera natural la zona sembrada cuando se identifique lluvia y protegerla al identificar granizo. Como el sensor identificará las precipitaciones cuando sucedan, la eficacia del dispositivo dependerá de la capacidad que tenga el algoritmo para comprobar que los datos tomados pertenecen al correspondiente fenómeno y, en el caso de que se implementara en un sistema mecánico de protección, a la velocidad de actuación de dicho sistema.

## II. PREPARACIÓN DE LA BASE DE DATOS

Para la creación de la base de datos se han buscado archivos de audios de lluvia, granizo y ruidos de diferentes tipos

en [www.freesound.org](http://www.freesound.org), ya que cuenta con licencia Creative Commons y su distribución es libre.

Debido a que tanto las frecuencias de muestreo como el formato de los audios no coincide de forma nativa, se ha recurrido a convertir todos los archivos a un formato y a una frecuencia de muestreo común. El formato utilizado ha sido WAV por codificación PCM con una resolución de 16 bits y una frecuencia de muestreo de 44100 Hz. Al no ser ninguna de las frecuencias de muestreo inferiores a esta no ha habido ningún problema con generaciones de ceros para rellenar el espectro. El programa utilizado ha sido *aTube Catcher*, ya que se trata de un programa con licencia gratuita.

Para dotar a MATLAB del repertorio de audios se ha creado el script *AudioImport*, que importa todos los archivos de audio y escoge un segmento de 5 segundos a partir de la mitad del audio. Se ha considerado esta medida debido a que algunos audios tienen una duración demasiado larga en comparación a la necesaria para el desarrollo del algoritmo. Además, se comprobó que en la mayoría se encuentra actividad relevante para el estudio a partir de la mitad del audio. En este script también se ha normalizado el volumen de todos los audios para que los valores de amplitud no difieran entre ellos.

Además de los audios de los sonidos requeridos para la distinción ha sido necesario añadir sonidos que no sean los anteriores, pero tengan unas características similares, ya que es necesario hacer una clasificación de lo que no es considerado lluvia o granizo para evitar errores a la hora de ejecutar el algoritmo. Como sonidos de características similares se han utilizado archivos con diferentes tipos de ruido, ya que se pueden llegar a confundir con las precipitaciones por el carácter ruidoso de estas.

## III. EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características es la parte más importante del proceso, y de ella dependerá el grado de acierto del algoritmo una vez que se ha probado. Las señales de lluvia y granizo se pueden considerar "Noise-Like signals" [2], ya que se componen de sonidos percusivos con mayor o menor distancia temporal entre ellos, según el volumen de lluvia o

granizo sea mayor o menor. Al tratarse de sonidos percusivos no existen componentes tonales, lo que ha dado lugar a acotar las características que se pueden elegir. Se ha utilizado una ventana rectangular de 1024 muestras y un solapamiento de 100 muestras, para tener en cuenta aproximadamente un 10% de la información de la ventana previa al analizar la característica deseada en cada ventana.

#### A. Short Time Energy

Se trata del único parámetro utilizado en el dominio del tiempo y representa la variación de amplitud de la señal en el tiempo. Se ha considerado un parámetro apropiado para la distinción entre granizo y lluvia ya que el granizo contiene golpes más percusivos y distinguidos que la lluvia, lo que en el tiempo da lugar a variaciones de amplitud más abruptas.

La Fig. 1. representa una lluvia fuerte y un granizo en el que apenas hay lluvia de fondo, pero al final del audio se distingue una caída de granizo acumulado.

$$U(n) = \sum_{m=-\infty}^{\infty} T[S(m)]W(n-m) \quad (1)$$

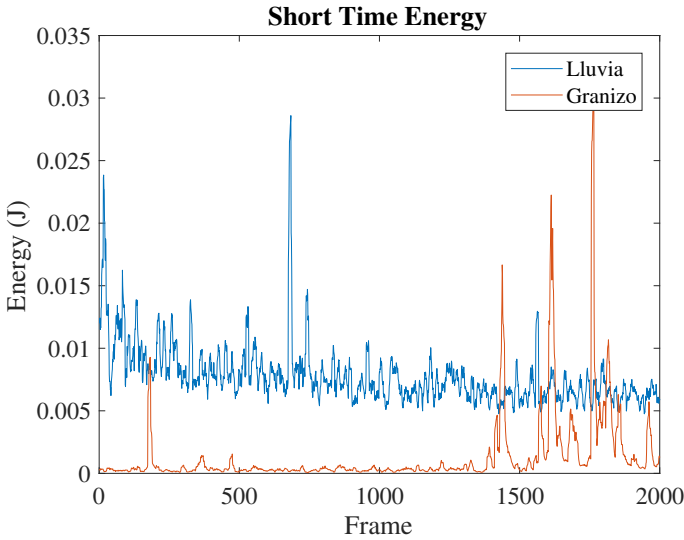


Fig. 1. Short Time Energy lluvia y granizo

#### B. Centroide Espectral (Spectral Centroid)

El Centroide Espectral es una característica que define el centro de gravedad del espectro, considerando a este como una distribución de probabilidad, determinada por la amplitud del espectro para cada frecuencia [5]. Se ha considerado de utilidad ya que el granizo presenta unos valores de amplitud en aquellas frecuencias que abarcan los golpes por la precipitación, y esto hará que el valor difiera con respecto al de la lluvia.

$$\begin{cases} \mu = \int x p(x) \delta x \\ p(x) = \frac{\text{amplitud}(x)}{\sum_x \text{amplitud}(x)} \end{cases} \quad (2)$$

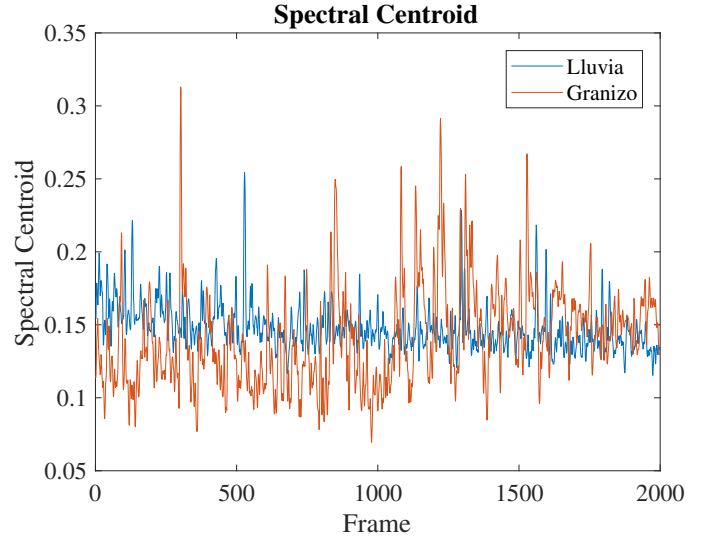


Fig. 2. Spectral Centroid lluvia y granizo

#### C. Variación Espectral (Spectral Flux)

La Variación Espectral aporta información sobre la variación del espectro que se produce entre dos frames consecutivos, y tiene utilidad a la hora de diferenciar entre música y sonidos del entorno [5].

$$flux(t) = 1 - \frac{\sum_k a(t-1, k) a(t, k)}{\sqrt{\sum_k a(t-1, k)^2} \cdot \sqrt{\sum_k a(t, k)^2}} \quad (3)$$

El parámetro tiende a 0 cuando hay gran similitud entre los espectros. Esta característica ha sido considerada necesaria para distinguir entre la lluvia y el granizo, ya que el granizo se puede producir con lluvia de fondo, lo que hace que haya en principio un espectro sin variación, pero a medida que se produzcan los sonidos del granizo haya cierta variación en el espectro.

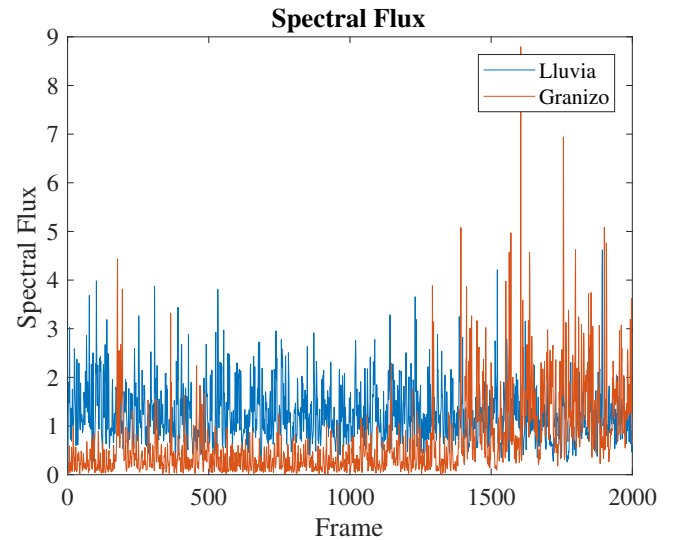


Fig. 3. Spectral Flux lluvia y granizo

En la Fig. 3 se observa que cuando cae el granizo aumenta el valor del Spectral Flux.

#### D. Roll-Off Espectral (Spectral Roll-Off)

El Roll-Off Espectral considera el espectro como una distribución de probabilidad. En términos espectrales, la característica es aquella frecuencia para la cual se tiene cierto porcentaje de la energía por debajo de la misma [5]. Para el desarrollo del algoritmo se ha considerado un 85% de la energía, que es un valor estándar de uso.

$$\sum_0^{fc} a^2(f) = 0.85 \sum_0^{fs/2} a^2(f) \quad (4)$$

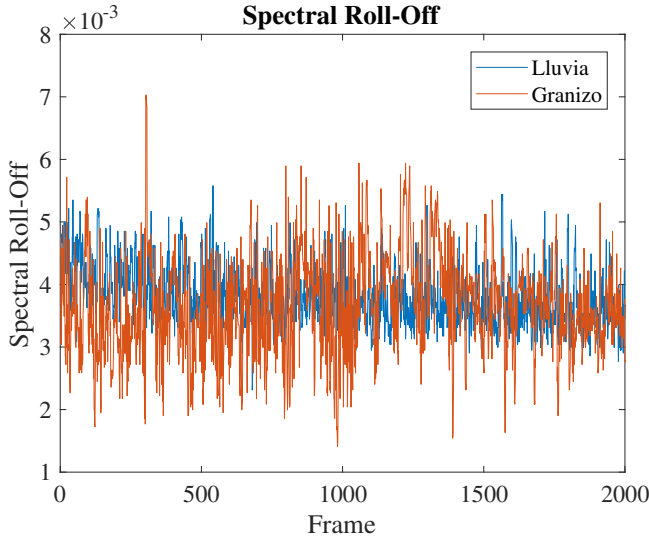


Fig. 4. Spectral Roll-Off lluvia y granizo

#### E. Densidad espectral de potencia

La Densidad espectral de potencia informa sobre la distribución de la potencia de una señal sobre las distintas frecuencias de las que está formada.

$$E = \int_{-\infty}^{\infty} Sxx(f)df \quad (5)$$

El golpe de las precipitaciones produce componentes espectrales dentro del rango audible. Con el cálculo de la densidad espectral de potencia en el rango de frecuencias en el que trabaja la lluvia será posible encontrar otro parámetro de distinción. En el caso de la lluvia la banda de frecuencias abarca los 600-1200 Hz [1]. No se trata de uno de los factores que más acentúe las diferencias entre el granizo y la lluvia, ya que muchas veces el propio granizo cae entre un fondo lluvioso, por lo que esas componentes existirán en ambos tipo de audio. Por otra parte puede ser un método de distinción para audios en los que solo se escuche el golpe del granizo, ya que actúa en una banda diferente.

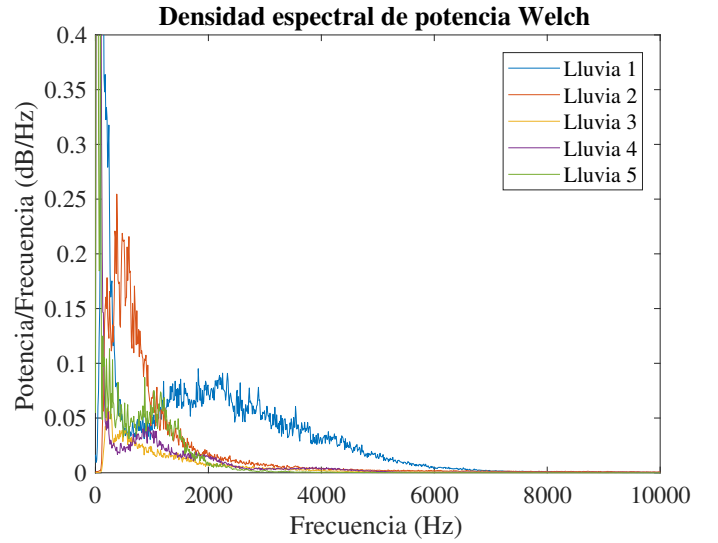


Fig. 5. Densidad espectral de potencia Welch Lluvia

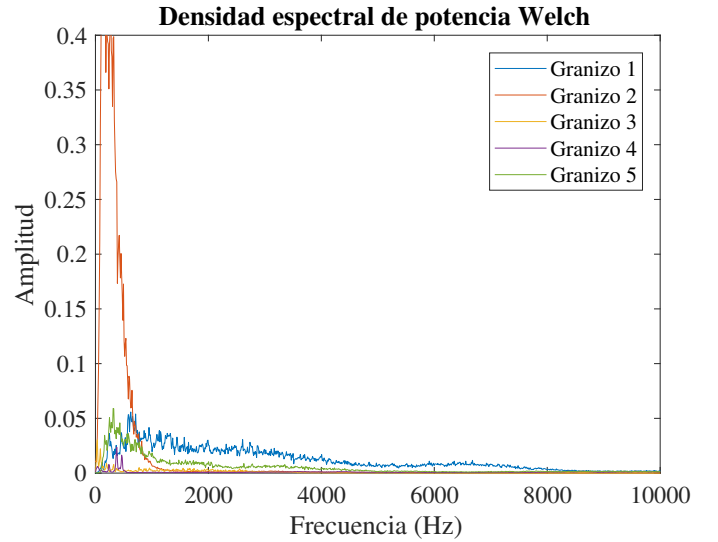


Fig. 6. Densidad espectral de potencia Welch Granizo

## IV. CLASIFICACIÓN

Una vez obtenidos los parámetros se ha diseñado la matriz de entrenamiento para ejecutar con la función Classify. El repertorio de archivos para realizar la comparación consta de 21 audios: 3 grupos de 7 audios, cada uno representando al suceso que se quiere distinguir (Lluvia, Granizo y desconocido). De cada uno se extraen los parámetros y se insertan en la matriz de entrenamiento.

En un principio se hicieron pruebas con la mecánica *Leave One Out* ejecutando el programa de manera manual para cada audio, lo cual requería una inversión de tiempo considerable. Por ello, se ha decidido crear el script *Trainer* con una interfaz *User-Friendly* permitiendo utilizar tres pruebas diferentes:

### A. New Samples

Realiza la clasificación con los archivos incluidos en la carpeta *Test*, destinada a ficheros ajenos a los de la matriz de entrenamiento, devolviendo el grupo al que el algoritmo considera que pertenece.

### B. Leave One Out

Se trata de un apartado para comprobar el funcionamiento de la matriz de entrenamiento utilizando los audios que la componen, de forma individual. El programa permite elegir uno de ellos y realizar su clasificación.

### C. Auto Test

Consiste en un procedimiento que realiza el Leave One Out con cada uno de los audios de la matriz de entrenamiento y devuelve aquellos audios que no se identifican con el grupo asignado. Además, devuelve el total de aciertos y fallos, así como el porcentaje de probabilidad de éxito. Es un método pensado para probar tanto la fidelidad de los archivos que se utilizan como los parámetros que se han escogido.

## V. TIEMPO DE COMPUTACIÓN

Para estimar el tiempo de computación se han utilizado dos procedimientos:

### A. Tiempo calculado en programa

De esta manera, al ejecutar cualquiera de las opciones que genera el *Trainer* devuelve el tiempo que ha tardado en ejecutarse el procedimiento al finalizar este, sin utilizar la herramienta *Profiler*. En este caso solo se devuelve el tiempo general de ejecución, prescindiendo del tiempo por ejecución dentro del código. En el caso de la función *NEW SAMPLES* se ha logrado un tiempo de ejecución de 4.13 segundos. En la función *LEAVE ONE OUT* se ha logrado un tiempo de 0.32 segundos. Finalmente, se ha logrado un tiempo de 5.87 segundos en el *AUTO-TEST*

### B. Tiempo calculado con la función Profiler

Para ello se ha ejecutado el script *Trainer* con la función *Run And Time*. El programa ha devuelto una tabla con numerosos valores que correspondían a la creación de la matriz de entrenamiento y a la presentación de la interfaz. Como MATLAB devuelve una tabla con demasiados valores se ha procedido a representar los 10 más significativos:

TABLE I  
TIEMPOS DE COMPUTACIÓN TRAINER

Function Name	Calls	Total Time
trainer	1	23.927 s
featurevector	34	11.896 s
new_sample	1	4.763 s
pwelch	68	4.215 s
signal\private\welch	68	4.209 s
signal\private\welch>localComputeSpectra	68	4.162 s
signal\private\computeperiodogram	15210	3.938 s
signal\private\computeDFT	15210	3.167 s
sig...ate\computeDFT>computeDFTviaFFT	15210	3.096 s
psdfreqvec	15278	2.870 s

Se puede observar que la mayor parte del tiempo se invierte en realizar la matriz de entrenamiento, concretamente el parámetro de la densidad espectral de potencia.

## VI. CONCLUSIÓN Y APLICACIONES FUTURAS

En vista de los resultados obtenidos, el algoritmo no debería ser utilizado en aplicaciones de seguridad de las que dependan vidas, ya que requiere una optimización para reducir el número de errores considerablemente. En la prueba del *AUTO-TEST* se ha obtenido una tasa de aciertos del 66,7%, con lo que el resultado es poco satisfactorio. En la prueba se han incluido varios archivos con sonidos de nevadas y el resultado indica que algunos se identifican como Ruido y otros como Granizo. Por otra parte, en la identificación de ficheros de lluvia el resultado no ha sido satisfactorio, ya que ha reconocido 1 de 5 audios de lluvia introducidos. Un inconveniente encontrado a la hora de implementar el algoritmo es que los espectros de los audios difieren entre sí a pesar de tratarse de la misma precipitación, lo que hará que, si se llega a utilizar con un sensor, sea muy dependiente de la posición en la que está colocado, lo que hará que el dispositivo sea ineficiente.

Como una parte importante de los sensores es su funcionamiento en tiempo real, un desarrollo futuro podría ser la implementación del algoritmo para que capture el audio y, procesándolo previamente, realice una comparación con la matriz de entrenamiento. Una forma de implementación podría consistir en una toma de 5 segundos cada cierto tiempo y la ejecución del programa, de esta manera tendríamos un funcionamiento similar al de este proyecto.

Para invernaderos de tamaño medio podría ser conveniente el diseño de un sistema de protección del cultivo utilizando este algoritmo con la finalidad de proteger la plantación en caso de granizo. El algoritmo podría estar implementado en un sistema con micrófonos que al instante activara un mecanismo que cubra la plantación.

## REFERENCES

- [1] Carol Bedoya, Claudia Isaza, Juan M. Daza, José D. López. *Automatic identification of rainfall in acoustic recordings*. Medellín, Colombia, 2016.
- [2] Alexander Brown, Saurabh Garg, and James Montgomery. *Automatic Rain and Cicada Chorus Filtering of Bird Acoustic Data*. Tasmania, Australia, 2018.
- [3] Seoane, Manuel A Sobreira and Molares, Alfonso Rodríguez. *Clasificación automática de fuentes de ruido de tráfico*. 2008.
- [4] Theodoros Giannakopoulos. *Some Basic Audio Features Matlab Code*. <http://www.di.uoa.gr/~tyiannak>, 2014.
- [5] Fabián Aguirre Martín. *Desarrollo y análisis de clasificadores de señales de audio*. Escuela Politécnica Superior de Gandía, 2014.
- [6] <http://www.anti-grele.fr>. *Los sistemas de protección granifugos*. 2016.
- [7] <https://agriculturers.com>. *El problema de la precipitación en la agricultura*. 2018.