# Supresión de ruido de teclas en videollamadas

*Catriel Bartezaghi*, *Esteban J. Odetti*, *Sebastián E. Scioli*
catrielbarte@gmail.com, estebanode@gmail.com, sebastianscioli@gmail.com
Procesamiento Digital de Imágenes – Departamento de Informática – FICH UNL

***Resumen*— La pandemia por COVID-19 acentuó el uso de la virtualidad haciendo que ésta se introduzca en casi todos los aspectos de la vida. Muchos trabajos e incluso clases comenzaron necesariamente a realizarse de manera virtual, lo cual trajo aparejado algunos problemas como ruidos molestos de fondo, entre los que se destaca el de las teclas. En este trabajo se propone eliminar el ruido impulsivo de un teclado presente en las videoconferencias. Los pasos básicos son detectar la presencia de teclas en el audio, eliminarlas y, por último, reconstruir el tramo eliminado. La mayor parte del tiempo se trabaja en el dominio tiempo-frecuencia. La salida del sistema es una señal en el dominio del tiempo sin la presencia del ruido producido por el tecleo.**

***Palabras claves*— teclado, ruido impulsivo, eliminación, grabaciones.**

## I. INTRODUCCIÓN

En reiteradas ocasiones, mientras las personas se encuentran en una videoconferencia los distintos participantes pueden llegar a utilizar el teclado mientras se encuentran hablando. Este ruido puede interferir en el discurso del interlocutor, sobreponiéndose a las palabras y siendo una molestia para los distintos participantes. Es por ello que en el presente trabajo se expone una implementación para reducir este problema.

El ruido del teclado se puede considerar impulsivo, por lo que es un caso puntual del mismo. Se implementaron algoritmos variados para poder suprimir el mismo del discurso del hablante: un algoritmo de detección, dos de eliminación, tres de reconstrucción y la herramienta de síntesis de Octave. Si bien algunos métodos de reconstrucción realizados en este trabajo otorgan buenos resultados, existen otros más complejos como por ejemplo el uso de inteligencia artificial para reconstruir lo eliminado con MAP [1], que se basa en métodos Bayesianos. Se utilizaron distintas señales de voz con ruido de teclado y se evaluaron los resultados usando las métricas PESQ y STOI.

## II. MATERIALES Y METODOLOGÍAS

En esta sección se describe cómo se generó el dataset de pruebas, variando la SNR, y además las características intrínsecas de los sonidos de teclas, que hacen posible su reconocimiento. El software base utilizado fue *Octave* v6.2.0 y *Audacity* para editar las señales.

### A. Generación de dataset

Para obtener el dataset a utilizar se realizaron diferentes pruebas en donde un participante hablaba y otras donde sólo sonaba el teclado, a lo largo de 5 segundos. Así se obtuvieron señales limpias (solo voz) y señales con solo ruido de teclado. La frecuencia de muestreo en todas las grabaciones fue de 44100 Hz, característica propia del micrófono que se utilizó. Para obtener la señal de voz contaminada con ruido de teclado se utilizó una relación señal-ruido de -20, 0 y 20 decibelios, con el cual se calculó un parámetro α (peso que multiplica al ruido) utilizando la definición de $SNR_{dB}$, quedando:

$$\alpha = \sqrt{\frac{Ps}{Pr * 10^{\frac{snr}{10}}}}$$

Donde *Pr* es la potencia de la señal limpia, *Ps* la potencia de la señal solo con ruido y *snr* es la relación señal ruido. De esta manera, la señal generada *Sr* queda:

$$Sr = S + \alpha * r$$

donde *S* es la señal de voz limpia y *r* la señal de ruido.

### B. Características del sonido de teclas

El ruido de las teclas se considera impulsivo y espectralmente plano, esto quiere decir que contiene todas las componentes frecuenciales. Al momento que un individuo presiona una tecla se producen dos ruidos: uno cuando se presiona y otro cuando se suelta la tecla, esto es por la naturaleza mecánica del teclado. Se sabe, gracias a otros estudios, que la duración de la tecla puede estar entre los 60 ms y 80 ms normalmente, pero en ocasiones puede

llegar hasta los 200 ms. Además, se considera aleatorio al ruido producido por el teclado y esto es por los diferentes tipos de teclados, diferentes mecanográficas, léase esta última como más fuerza al presionar la tecla, más velocidad, etc.

## III. ALGORITMO

A continuación se muestra el flujo completo del sistema, iniciando con la señal con ruido capturada y culminando con señal post procesamiento como salida:

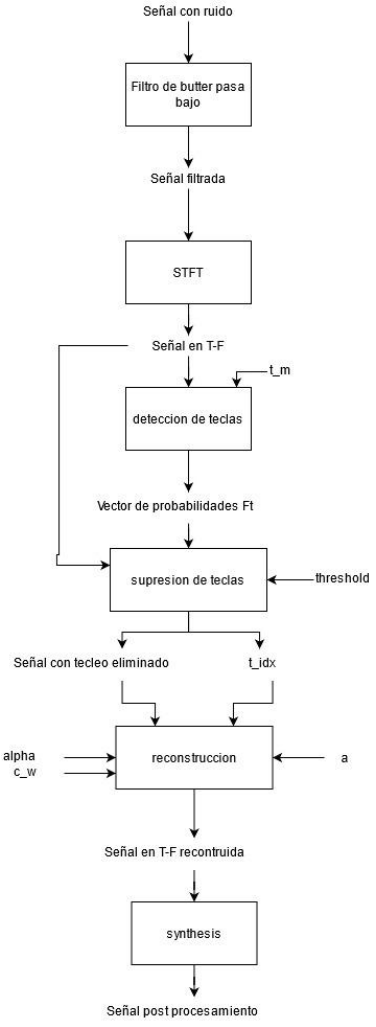


**Fig. 1**: Diagrama en bloque del algoritmo implementado.

### A. Preprocesamiento de señal de entrada

En este bloque ingresa la señal a procesar. En el procesamiento de voz las bajas frecuencias son las más importantes, por lo que se utilizó un filtro pasa bajo con frecuencia de corte de 8000 Hz. Se escogió Butterworth para aprovechar la poca ondulación en las bandas de paso y de rechazo, además se eligió un orden $n = 40$ relativamente alto para obtener una precisión acorde a lo que necesitaba el problema.

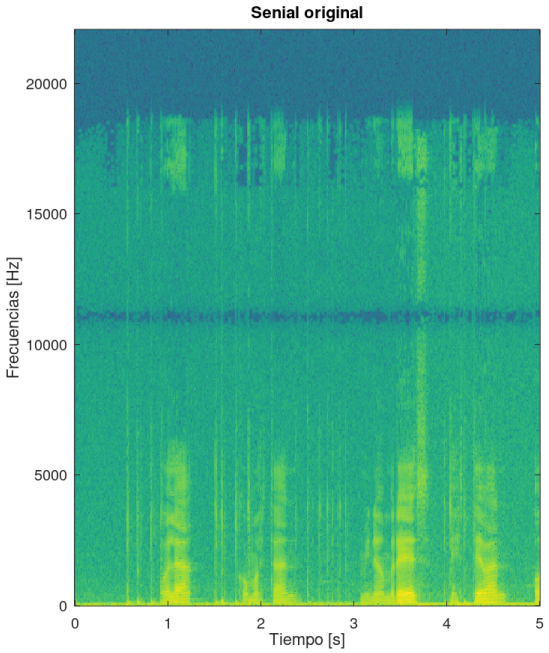


**Fig. 2**: Señal contaminada con ruido de teclado en el dominio de T-F antes de ser filtrada.

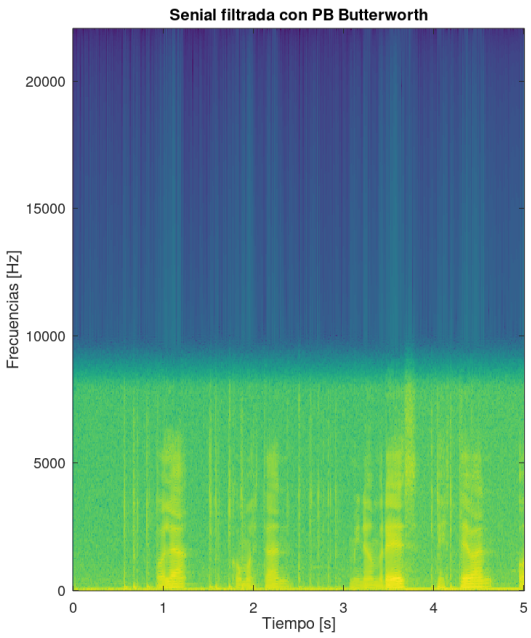


**Fig. 3**: Señal contaminada con ruido de teclado en el dominio de T-F después de ser filtrada.

En la figura 3 se puede observar el corte en 8000 Hz.

### B. STFT

La ventaja principal que ofrece el análisis en el dominio del tiempo y frecuencia es que otorga información sobre las frecuencias que hubo a lo largo del tiempo. Además es útil para analizar señales con parámetros variables en el tiempo, llamadas señales aleatorias no estacionarias. La

entrada a la STFT en este caso es la señal en el dominio del tiempo previamente filtrada, y la salida es una matriz en la que cada columna representa las ventanas temporales en donde está calculada la FFT de cada ventana. Se utilizó una ventana de Hanning por su amplia utilización en el procesamiento de la voz. El tamaño de la ventana fue de 20 ms y el porcentaje de solapamiento de las mismas se mantuvo fijo en 50%.

### C. Detección y supresión de teclas

Para la detección de teclas se utiliza un modelo de regresión, dado por:

$$S(k,t) = \sum_{m=1}^{M} (\alpha_{km} * S(k, t-\tau_m)) + V(k,t)$$

El mismo combina una cierta cantidad de ventanas temporales (es decir, columnas) y predice una determinada columna en base a los vecinos modelando gracias a una distribución normal con media

$$\sum_{m=1}^{M} \alpha_{km} * S(k, t-\tau_m)$$

y varianza

$$\sigma_{tk} = \frac{1}{M} * \sum_{m} (S(k, t-\tau_m))^2$$

Por lo tanto, se puede escribir:

$$p(S(k,t)|S(k,t-\tau_1), \dots, S(k,t-\tau_M)$$

$$=$$

$$N\left[\sum_{m=1}^{M} (\alpha_{km} * S(k,t-\tau_m)), \frac{1}{M} * \sum_{m=1}^{M} (S(k,t-\tau_m))^2\right]$$

Donde $\tau_m$ son las columnas vecinas, $k$ las frecuencias, $t$ los tiempos y $M$ la cantidad de vecinos a considerar. Suponiendo que las componentes frecuenciales son independientes en un frame dado, la siguiente fórmula calcula la probabilidad para un determinado tiempo como un producto de probabilidades:

$$p(S(t)) = \prod_{k} p(S(k,t))$$

Transformando esta multiplicación en una suma, aplicando logaritmo, da:

$$F_t = \sum_{k} \log\left(p(S(k,t)|S(k,t-\tau_1), \dots, S(k,t-\tau_M))\right)$$

Donde $F_t$ es proporcional a:

$$\propto -\frac{1}{2} * \sum_{k} \frac{1}{\sigma_{tk}} * (S(k,t) - \sum_{m=1}^{M} (\alpha_{km} * S(k,t-\tau_m)))^2$$

El vector resultante $F_t$ representa probabilidades en escala logarítmica. Si el frame se puede aproximar correctamente el sistema es suave continuando con una tendencia dada, pero cuando se introduce un ruido impulsivo (posiblemente una tecla) se introducen también componentes frecuenciales que no había antes por lo que va a intentar predecir y va a fallar.

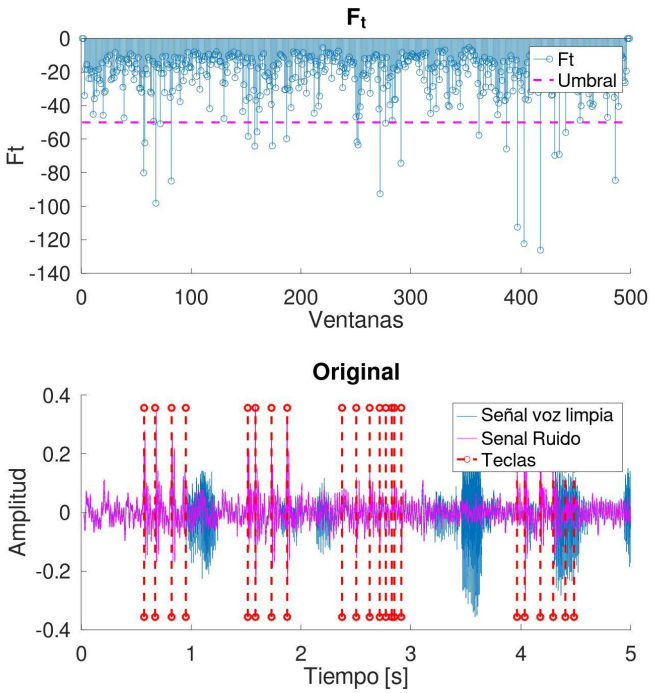


**Fig. 4**: De arriba hacia abajo: señal de voz contaminada con ruido de teclado con SNR cero y gráfica de su respectiva $F_t$.

Se puede observar en la figura 4 que cuanto más negativo es un valor de $F_t$ menos probabilidad de poder ser predicho con sus vecinos tiene un frame, siendo log(*1*) el valor máximo y log(*0*) el valor mínimo.

Si bien este método da resultados esperables, es muy dependiente del umbral que se utilice, por lo que puede dar falsos positivos o falsos negativos. En este trabajo se utilizaron umbrales variables desde -40 a -80.

Una vez detectados los frames contaminados con sonidos de teclas se los procede a eliminar completamente, asignando frecuencia cero en ese frame. Esto es posible

gracias a que el ruido que produce el teclado abarca todas las frecuencias.

### D. Métodos de reconstrucción de señal

Para la reconstrucción de la señal en las ventanas que fueron eliminadas se implementaron tres métodos, los cuales utilizan los valores vecinos para intentar estimar los valores más adecuados, y funcionan en el dominio de la STFT.

#### 1. Promedios Ponderados (PP)

En este caso se trata de un método no causal. La estimación de las ventanas faltantes se hace mediante un promedio ponderado de $n$ valores anteriores y posteriores correspondientes a la frecuencia de la ventana a calcular, y con un vector de pesos que decrece de acuerdo a la lejanía de la trama dañada, y que en total suma 1. El valor obtenido se reemplaza en todas las ventanas correspondientes a esa frecuencia.
El promedio ponderado queda definido por:

$$P = \sum_{n=1}^{c} ((c - n)/c\,((S(k, t - n)) + (S(k, t + n))))/2$$

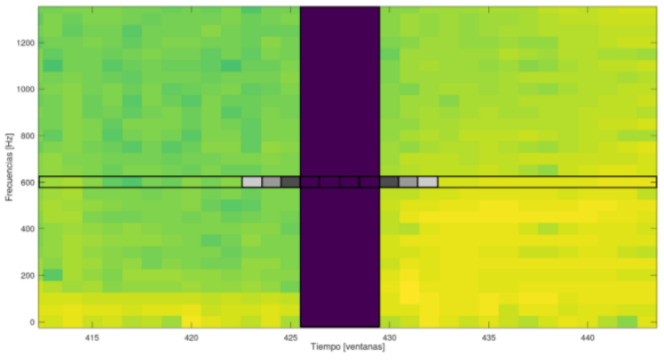


**Fig. 5**: Promedios ponderados. Pesos en escala de grises

#### 2. Interpolación Lineal (I)

Método no causal. Para la reconstrucción toma los dos valores más cercanos a la trama faltante (anterior y posterior) correspondientes al contenido frecuencial a reconstruir. Con estos 2 valores se realiza una interpolación lineal con la que se obtiene igual cantidad de valores a los faltantes. El cálculo está dado por:

$$I(n) = \sum_{n=1}^{c} (c - n)/c\,S(k, t) + (n/c)\,S(k, t + c)$$

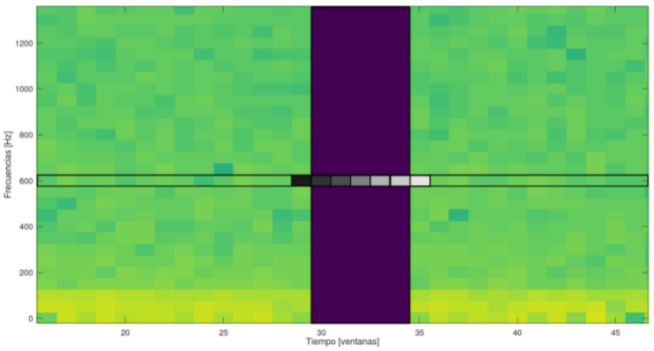


**Fig. 6**: Interpolación lineal. Pesos en escala de grises.

#### 3. Media Móvil (MM)

En este caso se el método puede ser causal, donde el cálculo se efectúa en base a ventanas anteriores, o no causal donde además se repite el cálculo para valores posteriores y se realiza un promedio. Para predecir el primer valor faltante se hace un promedio no ponderado con un rango $n$ de ventanas anteriores. Luego se repite para la ventana siguiente, corriendo el rango de ventanas de tal manera que incluya a la recientemente calculada. Se repite el proceso hasta obtener todos los valores. En caso del método no causal se repite el mismo cálculo pero de forma inversa con los valores posteriores.

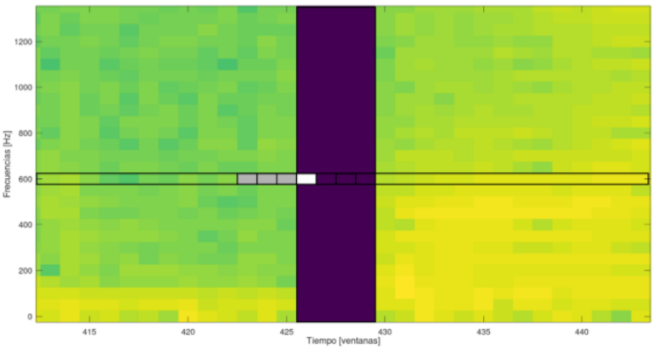


**Fig. 7**: Media móvil. Cálculo de ventana color blanco

## IV. EXPERIMENTOS Y RESULTADOS

### A. Parámetros utilizados

Para las pruebas que se llevaron a cabo se fueron variando distintos parámetros de manera de encontrar los adecuados para las grabaciones que se tienen, entre ellos la relación señal-ruido de -20 a 20 dB y el umbral de $F_t$ para detectar picos correspondientes a teclas, calculando para los tres métodos.

### B. Comparación de métodos de reconstrucción

Para comparar la calidad de la reconstrucción se utilizó la evaluación perceptual de la calidad del habla PESQ (ITU-T P.862) por sus siglas en inglés. Usualmente esta evaluación de resultados está entre 0 a 5 pero en este trabajo está normalizado entre 0 y 1 para facilitar la interpretación.

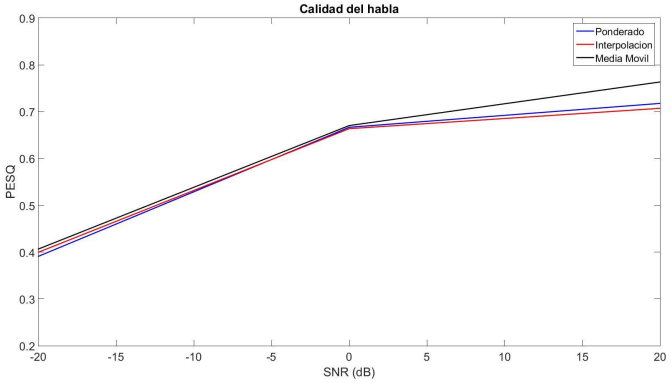


**Fig. 8**: Gráfica de PESQ con respecto a SNR.

Como se puede observar en la figura 5, el método que mejores resultados da es el de media móvil. Esto es lo esperado ya que el método es el que más vecinos al frame considera. También es esperable que a menos SNR peor sean los métodos ya que a medida que SNR disminuye, aumenta el ruido del teclado.

| SNR [dB] | Método 1 PP | Método 2 I | Método 3 MM |
|---|---|---|---|
| -20 | 0.391 | 0.399 | 0.407 |
| 0 | 0.667 | 0.664 | 0.670 |
| 20 | 0.718 | 0.707 | 0.763 |

**Tabla 1:** Valores de PESQ para métodos con SNR (dB)

Se puede observar que la tabla 1 confirma lo que gráficamente se ve en la figura 5. También se puede apreciar que en bajas SNR los métodos poseen resultados con poca variación, en cambio para altas SNR se ve claramente que el mejor método es el de MM. Cabe destacar que el método con menos valor en altas SRN es el de interpolación, pero en bajas SNR el método PP posee mejor puntaje.

## V. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se buscó usar señales de audio de videoconferencias para procesarlas eliminando el molesto sonido de fondo de teclas.

Los experimentos realizados para cada uno de los métodos de reconstrucción de la señal muestran resultados adecuados para los casos en los que SNR se encuentra alrededor de 0 dB, lo cual, perceptualmente, se considera que es similar al ruido real de teclas que puede aparecer en una videoconferencia.

Una posible mejora que podría hacerse es de alguna manera entrenar con más audios y combinaciones para encontrar un umbral de $F_t$ para aumentar la robustez del algoritmo.

También se podría implementar un método que obtenga información de las tramas dañadas, para mejorar la predicción. Se pueden implementar otros métodos, como por ejemplo interpolación cuadrática o splines.

Por último, podría implementarse en tiempo real el algoritmo, teniendo en cuenta algunas consideraciones técnicas como podrían ser capturar determinados segundos y guardarlos en un buffer para poder realizar el cálculo con mayor precisión.

## VI. REFERENCIAS

[1] A. Subramanya, M. L. Seltzer and A. Acero, "Automatic Removal of Typed Keystrokes From Speech Signals," in IEEE Signal Processing Letters, vol. 14, no. 5, pp. 363-366, May 2007, doi: 10.1109/LSP.2006.888091.

[2] Raj, B., Seltzer, M. L., & Stern, R. M. (2004). Reconstruction of missing features for robust speech recognition. Speech Communication, 43(4), 275–296. https://doi.org/10.1016/j.specom.2004.03.007

[3] Diego H. Milone Hugo, L. Rufiner Rubén C. Acevedo Leandro E Di Persia, Humberto M. Torres, *Introducción a las Señales y los Sistemas Discretos*, May 2009.