MEMORIA 2:

Pau Guàrdia i David Linde

1- Introducción :

En esta memoria hablamos de todos los procesos que hemos llevado a cabo para realizar la detección de voz de nuestro archivo de audio y de la base de datos proporcionada. También hablamos sobre las ampliaciones realizadas.
En primer lugar, hemos realizado el análisis de nuestro fichero de audio utilizando Wavesurfer y poniendo en un fichero .lab, en qué momento hay voz y en qué momento hay silencio.
Una vez realizado esto, hemos empezado con la configuración de nuestro entorno de trabajo y finalmente nos hemos puesto a programar.

2- Tareas y ejercicios

Tareas:

Para analizar la capacidad de un detector de voz basado en medidas estadísticas de la señal, como las calculadas en la primera práctica, utilice una señal de voz (16kHz, mono) que contenga pausas internas (puede utilizar la señal grabada en la primera práctica o grabar una nueva al efecto).

Nombre al fichero `pav_ggLD.wav` (gg: grupo 11, 41, etc; L: id. del pc; D: índice de señal, si graba más de una: 1, 2, ...). Los ficheros grabados por todos los alumnos del curso se utilizarán para evaluar los distintos sistemas de detección de voz.

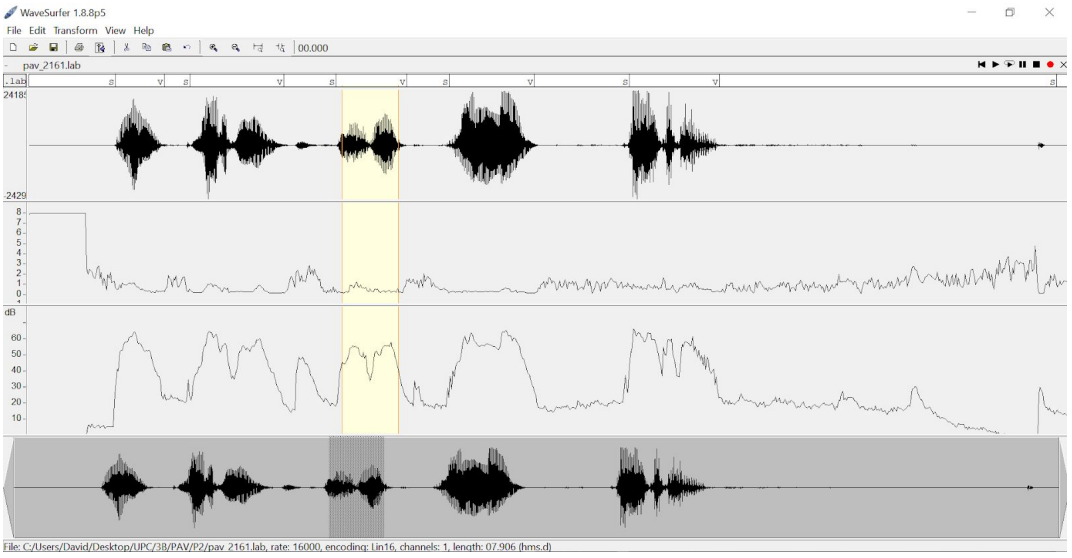
1. Tal y como hizo en la primera práctica, visualice en el programa `wavesurfer` la señal y las características potencia y tasa de cruces por cero. Cree un nuevo panel *transcription* y utilícelo para etiquetar los segmentos de silencio `(S)` y de voz `(V)`. Para ello, sitúese al final de cada segmento de voz o silencio e introduzca la etiqueta que corresponda. Sólo considere como silencio los segmentos de una cierta duración; no pausas cortas que no se perciban claramente.

   Guarde las transcripciones (fichero `.lab`) y suba los ficheros `.wav` y `.lab` a Atenea, para contribuir a la base de datos de evaluación.

2. Defina unas condiciones *razonables* para detectar la presencia de voz o silencio. Para ello, observe la señal y determine:

   - Nivel de potencia y tasa de cruces por cero en el silencio inicial de la señal.
   - Incremento aproximado del nivel de potencia, respecto al valor en el silencio inicial, que se tiene para distintos tipos de fonema: fricativas sordas, consonantes sonoras, vocales, etc.
     - ¿Es capaz de determinar un valor mínimo del incremento de nivel tal que, si se supera, podamos tener una cierta seguridad de que se trata de voz, pero, si no se hace, podamos considerar que se trata de silencio?
   - Duración mínima razonable de los segmentos de voz y de silencio.

3. Visualice con `less` o `cat` el fichero con la transcripción, e interprete el significado de su contenido.



El de arriba son los cruces por 0 y el de abajo power plot.

2.

Si observamos las gráficas anteriores, se observa que la que más variación contiene es la de la potencia. Se observa principalmente que al inicio de la grabación el nivel de potencia se encuentra muy bajo. Una vez se establece empieza lo que podríamos considerar silencio, que se encuentra ciertos dB por encima de este nivel esmentado.
Por otro lado se observa que siempre que hay voz el nivel de potencia se encuentra muy por encima al nivel de silencio, con lo que creemos oportuno que si esta 5dB por encima del nivel del silencio se podrá considerar ya que habrá voz en ese instante.

A partir del txt que generamos de la práctica anterior, hemos obtenido el ZCR del silencio inicia del .wav, basándonos en estos, no hemos podido destacar muchas condiciones, ya que no se visualizan gran cambios a lo largo del audio, y siempre se encuentran alrededor del nivel de silencio, unos fm/4. En nuestro audio no tenemos ninguna fricativa sorda, que nos harían observar un pico mucho mayor en este apartado.

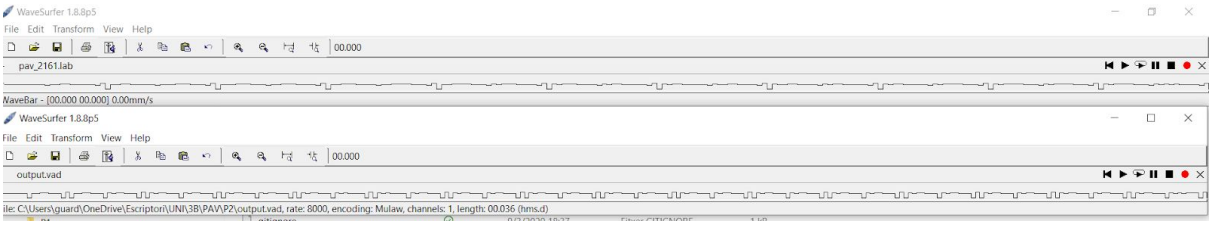
3.

```
pau@LAPTOP-5606KCNU:~/PAV/P2$ cat pav_2161.lab
0.0000000 0.6525000 S
0.6525000 1.4175000 V
1.4175000 2.5475000 S
2.5475000 3.4450000 V
3.4450000 4.5250000 S
4.5250000 5.7925000 V
5.7925000 6.8975000 S
6.8975000 7.8550000 V
7.8550000 8.5225000 S
8.5225000 9.3875000 V
9.3875000 10.7100000 S
10.7100000 10.9225000 V
10.9225000 11.1575000 S
pau@LAPTOP-5606KCNU:~/PAV/P2$
```

Podemos ver los tiempos entre los que hemos puesto que hay voz(V) y los que hay silencio(S).

Tareas:

- Ejecute `vad` con la señal que grabó y etiquetó al principio de la práctica.
- Abra la señal con `wavesurfer` y cree dos paneles de transcripción: uno con la segmentación manual (fichero `hola.lab`) y otro con la generada automáticamente (`hola.vad`).
- Compare el resultado obtenido con el resultado teórico.
  - Como el programa, en su estado actual, asigna un etiquetado aleatorio, no es de extrañar que no acierte ni una...



SCRIPTS/RUN_VAD.SH SCRIPTS/VAD_EVALUATION.PL

```
david@DESKTOP-4RUP236:~/PAV/P2$ scripts/run_vad.sh
**************** /home/david/PAV/P2/db.v4/2014/pav_4151.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4152.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4161.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4171.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4172.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4305.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4311.wav ****************
**************** /home/david/PAV/P2/db.v4/2014/pav_4341.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2109.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2110.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2151.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2152.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2171.wav ****************
**************** /home/david/PAV/P2/db.v4/2015/pav_2172.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_1341.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4111.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4112.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_41191.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_41192.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_412_1.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4131.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4132.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4151.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4161.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_418.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4311.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4312.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4313.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4321.wav ****************
**************** /home/david/PAV/P2/db.v4/201617q1/pav_4331.wav ****************
```

```
david@DESKTOP-4RUP236:~/PAV/P2$ scripts/vad_evaluation.pl db.v4/*/*lab
**************** db.v4/2014/pav_4151.lab ****************
Recall V:  4.85/6.94   69.88%   Precision V:  4.85/6.78   71.52%   F-score V (2)  : 70.20%
Recall S:  1.20/3.13   38.36%   Precision S:  1.20/3.29   36.51%   F-score S (1/2): 36.86%
===> db.v4/2014/pav_4151.lab: 50.869%

**************** db.v4/2014/pav_4152.lab ****************
Recall V: 22.32/28.13  79.34%   Precision V: 22.32/27.78  80.33%   F-score V (2)  : 79.54%
Recall S:  1.18/6.64   17.74%   Precision S:  1.18/6.99   16.87%   F-score S (1/2): 17.04%
===> db.v4/2014/pav_4152.lab: 36.809%

**************** db.v4/2014/pav_4161.lab ****************
Recall V:  2.76/3.09   89.25%   Precision V:  2.76/3.62   76.24%   F-score V (2)  : 86.30%
Recall S:  0.57/1.43   39.75%   Precision S:  0.57/0.90   63.06%   F-score S (1/2): 56.44%
===> db.v4/2014/pav_4161.lab: 69.790%

**************** db.v4/2014/pav_4171.lab ****************
Recall V:  3.77/5.02   75.07%   Precision V:  3.77/4.65   81.13%   F-score V (2)  : 76.21%
Recall S:  0.21/1.08   19.21%   Precision S:  0.21/1.46   14.28%   F-score S (1/2): 15.05%
===> db.v4/2014/pav_4171.lab: 33.868%

**************** db.v4/2014/pav_4172.lab ****************
Recall V:  4.24/5.57   76.17%   Precision V:  4.24/4.46   95.09%   F-score V (2)  : 79.33%
Recall S:  0.13/0.35   37.87%   Precision S:  0.13/1.46    9.14%   F-score S (1/2): 10.78%
===> db.v4/2014/pav_4172.lab: 29.238%

**************** db.v4/2014/pav_4305.lab ****************
Recall V:  2.37/3.06   77.34%   Precision V:  2.37/4.09   57.87%   F-score V (2)  : 72.46%
Recall S:  0.77/2.49   30.79%   Precision S:  0.77/1.46   52.51%   F-score S (1/2): 46.02%
===> db.v4/2014/pav_4305.lab: 57.747%

**************** db.v4/2014/pav_4311.lab ****************
```

```
**************** Summary ****************
Recall V:288.47/384.10 75.10%   Precision V:288.47/487.82 59.13%   F-score V (2)  : 71.25%
Recall S: 73.23/272.58 26.87%   Precision S: 73.23/168.86 43.37%   F-score S (1/2): 38.62%
===> TOTAL: 52.461%
```

Como podemos observar, al darnos un 52,46 % es totalmente aleatorio.

Ejercicios básicos:

1. Complete el código de los ficheros `main_vad.c` y `vad.c` para que el programa realice la detección de actividad vocal. Escriba las funciones de análisis o incorpore al proyecto los ficheros de la primera práctica `pav_analysis.c` y `pav_analysis.h`. Recuerde incorporar las cabeceras necesarias en los ficheros correspondientes.

   Tiene completa libertad para implementar el algoritmo del modo que considere más oportuno, pero el código proporcionado puede ser un buen punto de partida para hacerlo usando un autómata de estados finitos (FSA). Encontrará en los ficheros sugerencias para ello, marcadas con la palabra **`TODO`** (del inglés *to do*, a realizar).

   Ayúdese de la visualización de la señal y sus transcripciones usando `wavesurfer`, de la opción `-verbose` del programa `vad`, de la colocación de *chivatos* y de cualquier otra técnica que se le pueda ocurrir para conseguir que el programa funcione correctamente. Recuerde que el fichero de salida sólo debe incluir las etiquetas `V` y `S`.

VAD.C:

```
VAD_DATA *vad_open(float rate)
{
  VAD_DATA *vad_data = malloc(sizeof(VAD_DATA));
  vad_data->state = ST_INIT;
  vad_data->k0 = -100;
  vad_data->k1 = 0;
  vad_data->k2 = 0;
  vad_data->state_time = 0;
  vad_data->last_feature = 0;
  vad_data->sampling_rate = rate;
  vad_data->frame_length = rate * FRAME_TIME * 1e-3;
  return vad_data;
}
```

Al abrir el vad inicializamos todos los umbrales a 0 menos el k0 a -100 aparte de todos los tiempos y potencias a 0.

```
switch (vad_data->state)
{
case ST_INIT:
  if (vad_data->last_feature < f.p)
  {
    vad_data->k0 = f.p;
    if (f.p > vad_data->last_feature + 5)
    {
      vad_data->k0 = vad_data->last_feature;
      vad_data->k1 = vad_data->k0 + (f.p - vad_data->last_feature) - 2;
      vad_data->k2 = vad_data->k1 + 5;
      vad_data->state = ST_SILENCE;
    }
  }

  break;
```

En el primer estado, estado inicial, decidimos decidir como umbral k0 el máximo valor de potencia que obtenemos en las primeras tramas (cuando aún no hay silencio). Una vez se supera el valor inicial en 5 dB por lo que consideramos que ya se puede considerar que estamos en silencio, decidimos la k1 como la k0 anterior sumándole un cierto valor de dB que se caracteriza en la diferencia entre lo que ha aumentado la potencia respecto la trama anterior (es decir, la diferencia entre el nivel que había al establecerse el audio con el nivel que hay en el silencio). A parte se puede ver que le hemos restado 2 dB. Lo creemos

oportuno ya que no podemos asegurar que la primera trama de silencio se encuentra en el mínimo valor de potencia, pero como sí podemos asegurar que el nivel de potencia de silencio no varía más de 3 dB si le restamos 2 dB podremos asegurar que estamos detectando todo el silencio. Finalmente y observando la gráfica hemos decidido fijar un tercer umbral k2 que este 5 dB por encima del k1 donde ya podremos decidir que si se supera será voz.

```
case ST_SILENCE:
  Ninit = 0;
  if (f.p > vad_data->k1)
  {

    vad_data->state = ST_MaybeVOICE;
  }

  break;

case ST_MaybeVOICE:
  vad_data->state_time = vad_data->state_time + FRAME_TIME;
  if (f.p > vad_data->k1 && vad_data->state_time > 50)
  { /* Sponemos que el tiempo maximo de una pausa breve es de 50 ms*/
    vad_data->state = ST_SILENCE;
    vad_data->state_time = 0;
  }
  else if (f.p > vad_data->k2)
  {
    vad_data->state = ST_VOICE;
    vad_data->state_time = 0;
  }

  break;
```

Tal y como se explica en la práctica hemos decidido añadir dos estados más, que son el Maybe_SILENCE y el Maybe_VOICE. Estos dos se caracterizan en que si se supera cierto umbral estando en silencio o en voz decidiremos que se cambia de estado a estos dos. Una vez en estos dos si superamos el siguiente umbral será que ya podemos decidir que se pasa a voz o a silencio aunque si me quedo entre los dos, y pasa cierto tiempo que hemos fijado en 50 ms, nos quedaremos en el estado anterior. Principalmente lo que conseguimos con estos umbrales es que cuando haya una pausa corta propio de la respiración mientras hablas o incluso un ruido mientras hay silencio, estos no se detecten como silencio o voz respectivamente, pudiendo ser así mucho más realista la detección.

```
case ST_UNDEF:
  break;
}
vad_data->last_feature = f.p;
/*printf("%f %f %f\n", vad_data->k0, vad_data->k1, vad_data->k2);*/
if (vad_data->state == ST_SILENCE || vad_data->state == ST_VOICE)
  return vad_data->state;
else
  return ST_UNDEF;
```

Finalmente hemos decidido que la función vad solo devolverá el nombre del estado si este es silencio o voz, de otra forma te devolverá un estado indefinido (ST_UNDEF).

MAIN_VAD.C:

```
if (state != last_state)
{
  if (t != last_t)
  {
    if (last_state == ST_UNDEF && state == ST_SILENCE)
    {
      fprintf(vadfile, "%.5f\t%.5f\t%s\n", last_t * frame_duration, t * frame_duration, state2str(ST_SILENCE));
    }
    else if (last_state == ST_UNDEF && state == ST_VOICE)
    {
      fprintf(vadfile, "%.5f\t%.5f\t%s\n", last_t * frame_duration, t * frame_duration, state2str(ST_VOICE));
    }
    else
    {
      fprintf(vadfile, "%.5f\t%.5f\t%s\n", last_t * frame_duration, t * frame_duration, state2str(last_state));
    }
  }

  last_state = state;
  last_t = t;
}
```

Principalmente lo que cambiamos de este fichero es cómo se decide según el estado que me llega. Este se decidirá si el estado cambia pudiendo determinar así correctamente la duración del estado de voz o silencio. Obviamente si el estado que te llega es voz o silencio, se va a determinar que el segmento es voz o silencio. El problema será cuando te llegue un estado indefinido (será el caso de Maybe_VOICE, Maybe_SILENCE o ST_INIT). En estos casos lo que realizamos es decidir segun el próximo estado, es decir si el estado anterior a sido uno de los problemáticos, y el siguiente es silencio, se decidirá silencio. Cosa que provoca que si se había cambiado de silencio a un posible estado de voz pero después se vuelve a silencio, se considerará como si haya habido un ruido que te ha subido un poco la potencia, pero que este sigue siendo silencio. De la misma forma si hay una pausa breve con lo que se detecta un posible estado de silencio, pero el próximo estado es voz, se decidirá como voz ya que se considerará esta pausa breve. Con esto se considera solucionado el problema de las pausas breves y ruidos.

```
david@DESKTOP-4RUP2J6:~/PAV/P2$ scripts/vad_evaluation.pl pav_2161.lab
*************** pav_2161.lab ***************
Recall V:  4.96/4.96   99.90%   Precision V:  4.96/5.71   86.87%   F-score V (2)  : 96.99%
Recall S:  5.44/6.19   87.89%   Precision S:  5.44/5.45   99.91%   F-score S (1/2): 97.25%
===> pav_2161.lab: 97.118%
```

Observamos que con los umbrales decididos se consigue un 97,118% de detección de nuestro fichero de audio. Esto es lo máximo que hemos podido obtener ajustando todos los parámetros de los umbrales y de los tiempos.

2. Optimice los algoritmos y sus parámetros de manera que se maximice la puntuación de la detección de la base de datos de desarrollo (`db.v4`).

```
*************** db.v4/2018-19q1/pav_4382.lab ***************
Recall V:  7.03/7.21   97.48%   Precision V:  7.03/10.93  64.28%   F-score V (2)  : 88.35%
Recall S:  4.95/8.85   55.89%   Precision S:  4.95/5.13   96.45%   F-score S (1/2): 84.23%
===> db.v4/2018-19q1/pav_4382.lab: 86.265%

*************** db.v4/2018-19q1/pav_4385.lab ***************
Recall V:  2.48/3.24   76.46%   Precision V:  2.48/2.48   99.94%   F-score V (2)  : 80.23%
Recall S:  3.67/3.67   99.96%   Precision S:  3.67/4.43   82.77%   F-score S (1/2): 85.72%
===> db.v4/2018-19q1/pav_4385.lab: 82.929%

*************** Summary ***************
Recall V:338.44/384.10 88.11%   Precision V:338.44/397.63 85.11%   F-score V (2)  : 87.50%
Recall S:213.39/272.58 78.28%   Precision S:213.39/259.05 82.37%   F-score S (1/2): 81.52%
===> TOTAL: 84.456%
```

En caso de todos los audios el máximo que obtenemos justo corresponde con los mismos valores con los que hemos obtenido nuestro 100% y se obtiene un 84,456%. Si observamos un poco todos los ficheros de audio que hemos intentado detectar se observa que un gran número de ficheros se encuentran entre 85% y el 95%, el problema está en que algunos, pocos, se encuentran cerca del 60% y entonces la media baja mucho.
El problema de estos puede ser tal y como se explica en la práctica, que el umbral k0 de donde decimos los otros dos, no sea el más óptimos, donde señales con potencias bajas no se detectan correctamente ya que la diferencia entre segmentos de voz y silencio no son muy elevadas.

```
state = vad_close(vad_data);
/* TODO: what do you want to print, for last frames? */
if (t != last_t)
{
  if (state != ST_VOICE && state != ST_SILENCE)
  {
    fprintf(vadfile, "%.5f\t%.5f\t%s\n", last_t * frame_duration, t * frame_duration + n_read / (float)sf_info.samplerate, state2str(ST_SILENCE));
  }
  else
  {
    fprintf(vadfile, "%.5f\t%.5f\t%s\n", last_t * frame_duration, t * frame_duration + n_read / (float)sf_info.samplerate, state2str(state));
  }
}
```

Finalmente para la última trama, en caso de que el estado no sea ni voz ni silencio, hemos decidido que se decida silencio, ya que la mayoría de audios acaban en silencio.

Ejercicios de ampliación:

1. Complete el código del fichero `main_vad.c` para que el programa, en el caso de que se indique un fichero `.wav` de salida (opción `--output-wav`), escriba en él la señal de la entrada, sustituyendo por cero los valores de los intervalos identificados como silencio.

2. Modifique los ficheros necesarios para permitir que los parámetros del FSA (nivel de los umbrales, α_0, α_1...; duraciones mínimas y máximas; etc.) sean accesibles desde la línea de comandos (ver el Anexo II).

   Esta ampliación, de hecho, le puede resultar muy útil para optimizar el resultado de la detección.

```
if (sndfile_out != 0)
{
  sf_write_float(sndfile_out, buffer, frame_size);

  /* TODO: copy all the samples into sndfile_out */
}
```

Utilizamos la función sf_write_float() para escribir dentro del fichero sndfile_out el contenido de buffer, es decir el contenido del fichero de entrada.

```
if (sndfile_out != 0)
{
  if(state==ST_SILENCE){
    sf_write_float(sndfile_out, buffer_zeros,frame_size);
  }
  /* TODO: go back and write zeros in silence segments */
}
```

En este caso utilizamos la misma función pero lo que hacemos es copiar de buffer_zeros, un buffer lleno de ceros, consiguiendo así llenar el fichero de salida de ceros cuando nos encontramos en silencio.
Para verificar que funciona, hicimos a propósito, un ejemplo con umbrales mal elegidos, entonces, el fichero de audio generado era de muy mala calidad y de difícil entendimiento.
De esta forma, estábamos seguros de que el código hacia lo que queríamos.

```
pau@LAPTOP-5606KCNU:~/PAV/P2$ ninja -C bin
ninja: Entering directory `bin'
[2/2] Linking target vad.
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad output.wav
pau@LAPTOP-5606KCNU:~/PAV/P2$
pau@LAPTOP-5606KCNU:~/PAV/P2$ ls
MARKDOWN.md  db.v4      hola.lab  img          output.wav   pav_2161,lab.lab  pav_2161.vad  src
README.md    docopt_c   hola.txt  meson.build  p2_vad.pdf   pav_2161.lab      pav_2161.wav  zcr.txt
bin          ham.txt    hola.vad  output.vad   pav_216.lab  pav_2161.lab~     scripts
```

```
switch (vad_data->state)
{
case ST_INIT:
  if (vad_data->last_feature < f.p)
  {
    vad_data->k0 = f.p;
    if (f.p > vad_data->last_feature + 5)
    {
      printf("Introducir k0 (nivel de potencia primeras tramas): \n");
      scanf("%f", &vad_data->k0);
      vad_data->k1 = vad_data->k0 + (f.p - vad_data->last_feature) - 2;
      vad_data->k2 = vad_data->k1 + 5;
      vad_data->state = ST_SILENCE;
    }
  }

  break;
```

En este caso solo pediremos que nos entre por pantalla el primer umbral, ya que, como hemos explicado anteriormente los otros dos los calculamos respecto de este.

```
david@DESKTOP-4RUP2J6:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir automatico(0) o manual(1): 1
Introducir k0 (nivel de potencia primeras tramas):
-90
david@DESKTOP-4RUP2J6:~/PAV/P2$ scripts/vad_evaluation.pl pav_2161.lab
**************** pav_2161.lab ****************
Recall V:  4.87/4.96   98.04%   Precision V:  4.87/5.02   96.96%   F-score V (2)  : 97.82%
Recall S:  6.04/6.19   97.54%   Precision S:  6.04/6.14   98.41%   F-score S (1/2): 98.24%
===> pav_2161.lab: 98.027%
```

Como se observa si la k0 entrada es la correcta, en nuestro caso entorno a los -90 dB, se detectara de forma casi perfecta, 98% de efectividad. Quizá es el valor más correcto que se puede poner de k0 para nuestro fichero de audio, ya que poniendo este valor, obtenemos más precisión que con el sistema automático.

```
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir k0 (nivel de potencia primeras tramas):
5
pau@LAPTOP-5606KCNU:~/PAV/P2$ scripts/vad_evaluation.pl pav_2161.vad
**************** pav_2161.vad ****************
Recall V:  0.00/0.00    0.00%   Precision V:  0.00/0.00    0.00%   F-score V (2)  :  0.00%
Recall S: 11.23/11.23 100.00%   Precision S: 11.23/11.23 100.00%   F-score S (1/2):100.00%
===> pav_2161.vad: 0.000%

pau@LAPTOP-5606KCNU:~/PAV/P2$
```

Por otro lado si la k0 que te entran no tiene sentido con tu audio, se observa que detectas un 0%. Ya que en nuestro fichero de audio, al poner una k0 de 5, todos los umbrales están muy arriba y por lo tanto, todo es silencio según el programa.

3- Anexo:

Como ampliación nuestra hemos decidido incluir que el usuario pueda decidir la generación de un umbral automático (poniendo un 0) o que lo entre manualmente (1).

```
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir automatico(0) o manual(1): 0
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir automatico(0) o manual(1): 1
Introducir k0 (nivel de potencia primeras tramas):
-90
pau@LAPTOP-5606KCNU:~/PAV/P2$
```

```
case ST_INIT:
  if (vad_data->last_feature < f.p)
  {
    vad_data->k0 = f.p;
    if (f.p > vad_data->last_feature + 5)
    {
      printf("Introducir automatico(0) o manual(1): ");
      scanf("%d", &automatic);
      while (automatic!=0 && automatic!=1){
        printf("Error al introducir, por favor introduce 0 (automatico) o 1(manual):");
        scanf("%d", &automatic);
      }
      if (automatic == 0)
      {
        vad_data->k0 = vad_data->last_feature;
      }
      else
      {
        printf("Introducir k0 (nivel de potencia primeras tramas): \n");
        scanf("%f", &vad_data->k0);
      }
    vad_data->k1 = vad_data->k0 + (f.p - vad_data->last_feature) - 2;
    vad_data->k2 = vad_data->k1 + 5;
    vad_data->state = ST_SILENCE;
    }
   }

  break;
```

A parte como se observa, nos hemos asegurado que en el caso de que el usuario se equivoque al teclear, si el número introducido no es ni 0 ni 1, salta un mensaje de error para que se introduzca un número nuevo y así el programa no entra en colapso.

```
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir automatico(0) o manual(1): 5
Error al introducir, por favor introduce 0 (automatico) o 1(manual):4
Error al introducir, por favor introduce 0 (automatico) o 1(manual):6
Error al introducir, por favor introduce 0 (automatico) o 1(manual):1
Introducir k0 (nivel de potencia primeras tramas):
2
pau@LAPTOP-5606KCNU:~/PAV/P2$ bin/vad -i pav_2161.wav -o pav_2161.vad
Introducir automatico(0) o manual(1): 5
Error al introducir, por favor introduce 0 (automatico) o 1(manual):6
Error al introducir, por favor introduce 0 (automatico) o 1(manual):7
Error al introducir, por favor introduce 0 (automatico) o 1(manual):8
Error al introducir, por favor introduce 0 (automatico) o 1(manual):9
Error al introducir, por favor introduce 0 (automatico) o 1(manual):2
Error al introducir, por favor introduce 0 (automatico) o 1(manual):3
Error al introducir, por favor introduce 0 (automatico) o 1(manual):4
Error al introducir, por favor introduce 0 (automatico) o 1(manual):10
Error al introducir, por favor introduce 0 (automatico) o 1(manual):0
pau@LAPTOP-5606KCNU:~/PAV/P2$
```

4- Conclusiones :

Podemos decir que nuestro archivo de audio lo hemos optimizado al máximo consiguiendo de forma automática un 97,1 %. Lo cual es imperceptible si rellenamos el silencio en forma de ceros en un archivo de audio generado por nuestro código.
Por otra parte, en la base de datos, hemos obtenido un 84,4%.
En algunos casos, obtenemos un porcentaje mayor del 90% y en otros, un poco inferior del 80%. Aunque la mayoría de casos, los resultados estaban alrededor del 85 %.