ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ - ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
«ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»

ΜΑΘΗΜΑ : ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΩΝ ΦΩΝΗΣ ΚΑΙ ΗΧΟΥ

«Θέμα: Ανιχνευτής Ομιλίας»

ΑΡΚΟΛΑΚΗΣ ΔΗΜΗΤΡΙΟΣ
ΜΠΣΠ 13010

Πειραιεύς, Δεκέμβριος 2014

Εισαγωγή

Ο αλγόριθμος που υλοποιείται, υπολογίζει με αξιόπιστο τρόπο την αρχή και το τέλος ενός σήματος φωνής που περιέχει ανθρώπινη ομιλία. Τονίζουμε ότι θεωρούμε ότι τα πρώτα 100msec του σήματος δεν περιέχουν καθόλου ομιλία.

Το πρόγραμμα δέχεται ως όρισμα ένα αρχείο ήχου ομιλίας (μορφής wav, μονοκάναλο) και επιστρέφει τις παρακάτω τιμές:

B1: Η πρώτη εκτίμηση του πλαισίου στο οποίο ξεκινάει η ομιλία.
E1: Η πρώτη εκτίμηση του τέλους της ομιλίας.
B2: Τελική εκτίμηση του πλαισίου στο οποίο ξεκινάει η ομιλία.
E2: Τελική εκτίμηση του τέλους της ομιλίας.

Ορίζουμε ρυθμό δειγματοληψίας 10KHz. Το μέγεθος πλαισίου είναι 40 msec και η ολίσθηση πλαισίων 10 msec. Σύμφωνα με τα παραπάνω, έχουμε τις παρακάτω τιμές στο πρόγραμμα:

```
Fs=10000;       (ρυθμός δειγματοληψίας 10KHz)
winl=40*10^-3; (μέγεθος πλαισίου 40 msec)
wins=10*10^-3; (ολίσθηση πλαισίων 10 msec)
```

Το L υπολογίζεται ως εξής: “μέγεθος πλαισίου * ρυθμός δειγματοληψίας” οπότε στον παρόν αλγόριθμο παίρνει την τιμή των 400 δειγμάτων (0.04 * 10000). Παρομοίως, το R υπολογίζεται ως “ολίσθηση πλαισίων * ρυθμός δειγματοληψίας” και παίρνει την τιμή των 100 δειγμάτων (0.01 * 10000). Το “round” μας βοηθάει στην στρογγυλοποίηση των αριθμών, διότι L και R τα θέλουμε ως ακέραιες τιμές.

```
L=round(winl*Fs);
R=round(wins*Fs);
w=hamming(L);
Lx=length(x);
```

Υπολογισμός παραμέτρων βραχέος χρόνου

Μας δίνεται, ότι η ενέργεια βραχέος χρόνου, υπολογίζεται με τους παρακάτω τύπους:

$$E_r = \Sigma^{L-1}_{m=0} (s[rR+m]w[m])^2$$
$$\hat{E}_r = 10\log_{10}E_r - \max_r(10\log_{10}E_r)$$

Το Er υπολογίζεται για κάθε πλαίσιο r του σήματος, οπότε μπαίνει σε βρόγχο επανάληψης.

```
Er=[];
i=1;
while (i+L-1<=Lx)
   Er=[Er sum((x(i:i+L-1).*w).^2)];
   i=i+R;
end
Er=10*log10(Er)-max(10*log10(Er));
```

Μας δίνεται, ότι ο ρυθμός διέλευσης από το μηδέν, υπολογίζεται με τον παρακάτω τύπο:

$Z_r = R/(2L)\Sigma^{L-1}_{m=0}|sgn(s(rR+m)-sgn(rR+m-1)|$

Το Zr υπολογίζεται για κάθε πλαίσιο r του σήματος, οπότε μπαίνει σε βρόγχο επανάληψης.

```
Zr=[];
i=1;
while (i+L-1<=Lx)
   tmp=x(i:i+L-1);
   sumtmp=0;
   for k=2:L
      if (tmp(k)>=0 && tmp(k-1)<0) || (tmp(k)<0 && tmp(k-1)>=0)
         sumtmp=sumtmp+2;
      end
   end
   Zr=[Zr (R/(2*L))*sumtmp];
   i=i+R;
end
```

Κατώφλια

Όπως έχουμε ήδη τονίζει, θεωρούμε ότι τα πρώτα 100msec του ηχητικού σήματος δεν περιέχουν καθόλου ομιλία (λόγω του φυσιολογικού χρόνου αντίδρασης του ομιλητή από την έναρξη της ηχογράφησης). Οπότε τα αμελητέα αυτά πλαίσια υπολογίζονται ως εξής: “0.1 δευτερόλεπτα / ολίσθηση πλαισίου” (ακέραια τιμή, μέσω της στρογγυλοποίησης round).

```
fframes=round(.1/wins);
```

Για τον υπολογισμό του κατωφλίου ρυθμού διέλευσης από το μηδέν (IZCT) χρειαζόμαστε τις μέσες τιμές και τις τυπικές αποκλίσεις του λογαρίθμου της ενέργειας βραχέος χρόνου και του ρυθμού διέλευσης από το μηδέν, μέσα σε αυτό το διάστημα των 0,1sec.

```
eavg=mean(Er(1:fframes));     (μέση τιμή λογαρίθμου ενέργειας)
esig=std(Er(1:fframes));      (τυπική απόκλιση λογαρίθμου ενέργειας)
zcavg=mean(Zr(1:fframes));    (μέση τιμή ρυθμού διέλευσης από το μηδέν)
zcsig=std(Zr(1:fframes));     (τυπική απόκλιση ρυθμού διέλευσης από το μηδέν)
```

Το IZCT υπολογίζεται από τον παρακάτω τύπο, στον οποίο το IF είναι γενικό κατώφλι ανίχνευσης των μη έμφωνων πλαισίων, με σταθερά τιμή 35:

$IZCT = max(IF, zcavg + 3*zcsig)$

Οπότε, στο πρόγραμμα μας, έχουμε:

```
IF=35;
IZCT=max([IF zcavg+3*zcsig]);
```

Το ζεύγος κατωφλίων ITU και ITR χρησιμεύουν για την μέτρηση του λογαρίθμου της ενέργειας. Το ITU είναι μία σταθερά στο διάστημα [-10 έως -20]dB, για τις ανάγκες της άσκησης επιλέχθηκε η τιμή -15. Το ITR υπολογίζεται ως εξής:

*ITR = max(ITU-10, eavg + 3*esig)*

Οπότε, στο πρόγραμμα μας, έχουμε:

```
ITU=-15;
ITR=max([ITU-10 eavg+3*esig]);
```

Τα 5 βήματα!

Η αρχική αναζήτηση της περιοχής όπου η καμπύλη του λογαρίθμου της ενέργειας συγκεντρώνεται γύρω από ένα ενεργειακό μέγιστο, επιτυγχάνεται από μία ακολουθία πέντε βημάτων.

Στο πρώτο βήμα ο αλγόριθμος βρίσκει την τιμή του B1 (βλέπε αρχή εγγράφου). Η αναζήτηση ξεκινάει από το πρώτο πλαίσιο μέχρι να βρει πλαίσιο στο οποίο Er > ITR . Πρακτικά, το σημείο αυτό θεωρεί ότι είναι το σημείο όπου σταματάει ο θόρυβος (και η σιωπή) και ξεκινάει η ομιλία. Βεβαίως, γίνεται και μια αναζήτηση στα γύρω πλαίσια για να σιγουρευτεί ότι οι τιμές είναι Er > ITU και να επιβεβαιώσει ότι πρόκειται για αρχή συνεχούς ομιλίας και όχι για κάποιον άσχετο στιγμιαίο ήχο κλπ.

```
flag=1;
c=1;
B1=1;
while (flag)
   while (Er(c)<=ITR)
      c=c+1;
   end
   B1=c;
   flag=0;
   for c=B1+1:B1+3
      if c>Le
         break;
      end
      if Er(c)<ITU
         flag=1;
         break;
      end
   end
   if flag
      c=B1+1;
   else
      break;
   end
end
```

Στο δεύτερο βήμα με την ακριβώς ίδια λογική του πρώτου βήματος, αλλά με την αντίστροφη διαδικασία, βρίσκεται η πρώτη εκτίμηση του τέλους της ομιλίας E1. (βλέπε και στον κώδικα c=c-1, c=E1-1 και λοιπές διαδικασίες αντίστροφης αναζήτησης).

```
flag=1;
c=length(Er);
E1=c;
while (flag)
   while (Er(c)<=ITR)
      c=c-1;
   end
   E1=c;
   flag=0;
   for c=E1+1:-1:E1-3
      if c>length(Le)
         break;
      end
      if Er(c)<ITU
         flag=1;
         break;
      end
   end
   if flag
      c=E1-1;
   else
      break;
   end
end
```

Στο τρίτο βήμα, ο αλγόριθμος προσαρμόζει το κατώφλι ρυθμού διέλευσης από το μηδέν IZCT και θέτει έτσι το B2 το οποίο είναι και η τελική εκτίμηση για το σημείο αρχής της ομιλίας. Η αναζήτηση κοιτάει από το πρώτο σημείο εκτίμησης αρχής ομιλίας B1 και 25 πλαίσια πίσω, αν υπάρχει κάποιο πλαίσιο-πλαίσια όπου Zr > IZTC. Αν τα πλαίσια αυτά είναι λιγότερα από 4, το πρόγραμμα θεωρεί ότι δεν πρόκειται για ομιλία, οπότε δεν αλλάζει και το σημείο εκτίμησης αρχής ομιλίας (δηλαδή το αφήνει B2=B1). Αν όμως τα πλαίσια είναι από 4 και πάνω, θεωρεί ότι πρόκειται για ομιλία την οποία δεν “αντιλήφθηκε” η αναζήτηση του πρώτου βήματος με τα κατώφλια ITU και ITR, οπότε και ορίζει το B2 ως διαφορετικό του B1.

```
for i=B1:-1:B1-25
   if i<1
      break;
   end
   sumZ=0;
   ind=[];
   if Zr(i)>IZCT
      sumZ=sumZ+1;
      ind=[i ind];
   end
end
if sumZ>=4
   B2=ind(1);
else
   B2=B1;
end
```

Στο τέταρτο βήμα με την ακριβώς ίδια λογική του τρίτου βήματος, βρίσκεται η τελική εκτίμηση του τέλους της ομιλίας E2. Η αναζήτηση κοιτάει από το αρχικό E1 και 25 πλαίσια μετά και εφόσον τα πλαίσια είναι από 4 και πάνω (όπως και στο τρίτο βήμα) θεωρεί ότι πρόκειται για ομιλία την οποία δεν “αντιλήφθηκε” η αναζήτηση του τρίτου βήματος, οπότε και ορίζει το E2 ως διαφορετικό του E1 (αλλιώς το αφήνει E2=E1).

```
for i=E1:E1+25
    if i>Le
        break;
    end
    sumZ=0;
    ind=[];
    if Zr(i)>IZCT
        sumZ=sumZ+1;
        ind=[ind i];
    end
end
if sumZ>=4
    E2=ind(end);
else
    E2=E1;
end
```

Στο τελικό βήμα γίνεται και ο τελικός έλεγχος στην “καθαρή” ομιλία που έχουμε βρει μέχρι στιγμής. Σε περίπτωση που βρεθεί πλαίσιο όπου Er > ITR (σπάνιο για αρχεία ήχου ομιλίας) ο αλγόριθμος προσαρμόζει το B2 ή το E2.

```
i=B2-1;
while (1)
    if i<1
        i=1;
        break;
    end
    if Er(i)>ITR
        B2=i;
    else
        break;
    end
    i=i-1;
end

i=E2+1;
while(1)
    if i>Le
        i=Le;
        break;
    end
    if Er(i)>ITR
        E2=i;
    else
        break;
    end
    i=i+1;
end
```

Παρακάτω, βλέπουμε τον τελικό υπολογισμό των τιμών που μας επιστρέφονται, καθώς και την “εκτύπωση” τους. (πολλαπλασιάζουμε με ολίσθηση πλαισίου για να μας επιστρεφεί το σημείο σε δευτερόλεπτα)

```
E1 = E1 * wins;
E2 = E2 * wins;
B1 = B1 * wins;
B2 = B2 * wins;
```

B1
E1
B2
E2

Ο αλγόριθμος σε εφαρμογή

Παρακάτω παρατίθενται ορισμένα παραδείγματα χρήσης του αλγορίθμου, συνοδευόμενα από τα αποτελέσματα που επέστρεψαν. Για τις ανάγκες των παραδειγμάτων ηχογραφήθηκαν τρία διαφορετικά ηχητικά δείγματα:

greek.wav (οι αριθμοί 1 – 10 στα Ελληνικά)
english.wav (οι αριθμοί 1-20 στα Αγγλικά, με ένα μικρό επιτηδευμένο κενό ενδιάμεσα)
spanish.wav (οι αριθμοί 1-20 στα Ισπανικά, με ένα επιτηδευμένο μακρινό "βήξιμο" κάποια δευτερόλεπτα πριν ξεκινήσει η "καθαρή" ομιλία)

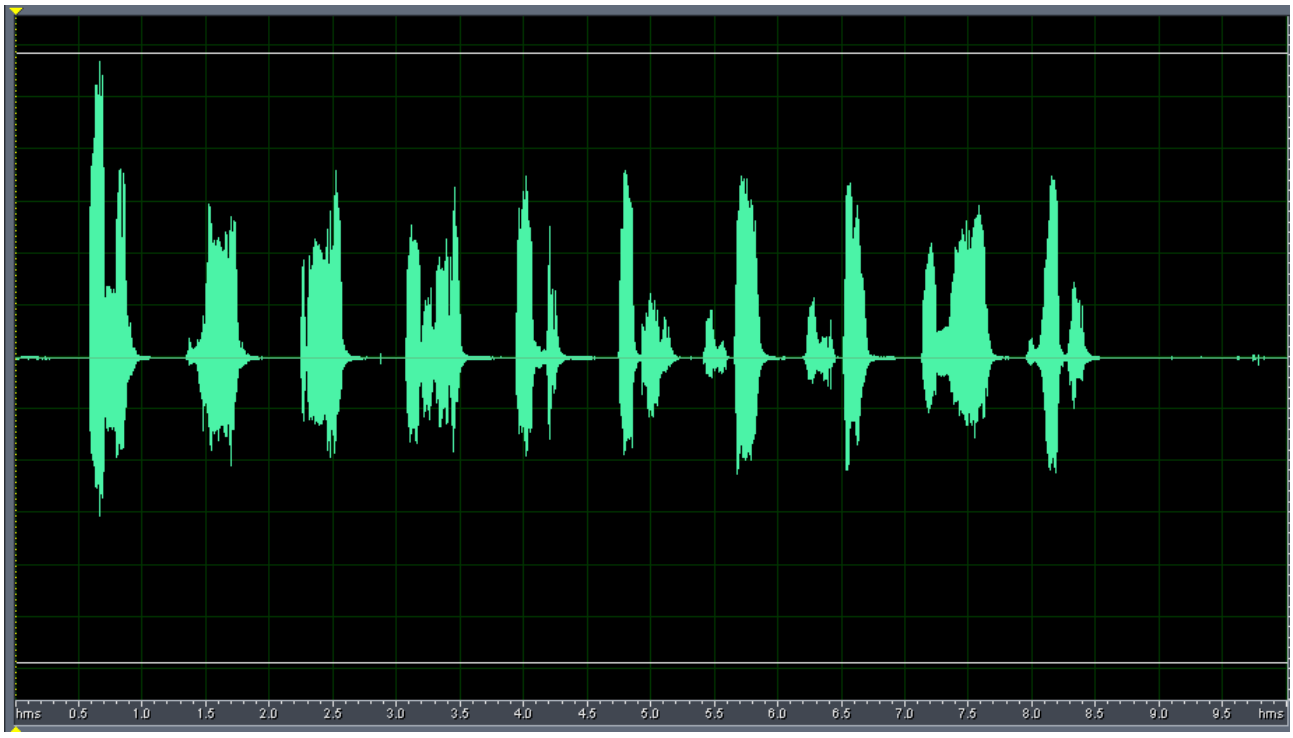
Για να εκτελέσουμε το πρόγραμμα, διαβάζοντας το "greek.wav", πρέπει να το τρέξουμε στην πλατφόρμα του matlab βάζοντας το αρχείο ως όρισμα, οπότε πληκτρολογούμε:

anixneyths_omilias('greek.wav')

Μας επιστρέφονται, λοιπόν, τα παρακάτω αποτελέσματα:

B1 = 0.5700
E1 = 8.3800
B2 = 0.5700
E2 = 8.3800

Στην παρακάτω εικόνα (αποτύπωση του αρχείου ήχου) βλέπουμε ότι όντως η αρχή και το τέλος της ομιλίας συμπίπτει με τα παραπάνω σημεία (βλέπε χρόνους στην κάτω μπάρα).



Παρομοίως, εκτελούμε το πρόγραμμα και για το αρχείο ήχου “english.wav”. Και παρατηρούμε ότι το κενό στη μέση δεν “μπερδεύει” το πρόγραμμα και βρίσκει την αρχή και το τέλος της ομιλίας κανονικά! Τα αποτελέσματα είναι τα εξής:

anixneyths_omilias('english.wav')

B1 = 0.9300
E1 = 17.3800
B2 = 0.9100
E2 = 17.3800

Τέλος εκτελούμε το πρόγραμμα και για το αρχείο “spanish.wav” και βλέπουμε ότι δεν λαμβάνει υπόψιν, ως μέρος της ομιλίας, το αρχικό “βήξιμο” που ακούγεται από μακριά στο δεύτερο δευτερόλεπτο περίπου, καθώς αντιλαμβάνεται ότι η ομιλία ξεκινάει προς το τέλος του έκτου δευτερολέπτου.

anixneyths_omilias('spanish.wav')

B1 = 6.9600
E1 = 22.9500
B2 = 6.9600
E2 = 22.9500


hms 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 hms