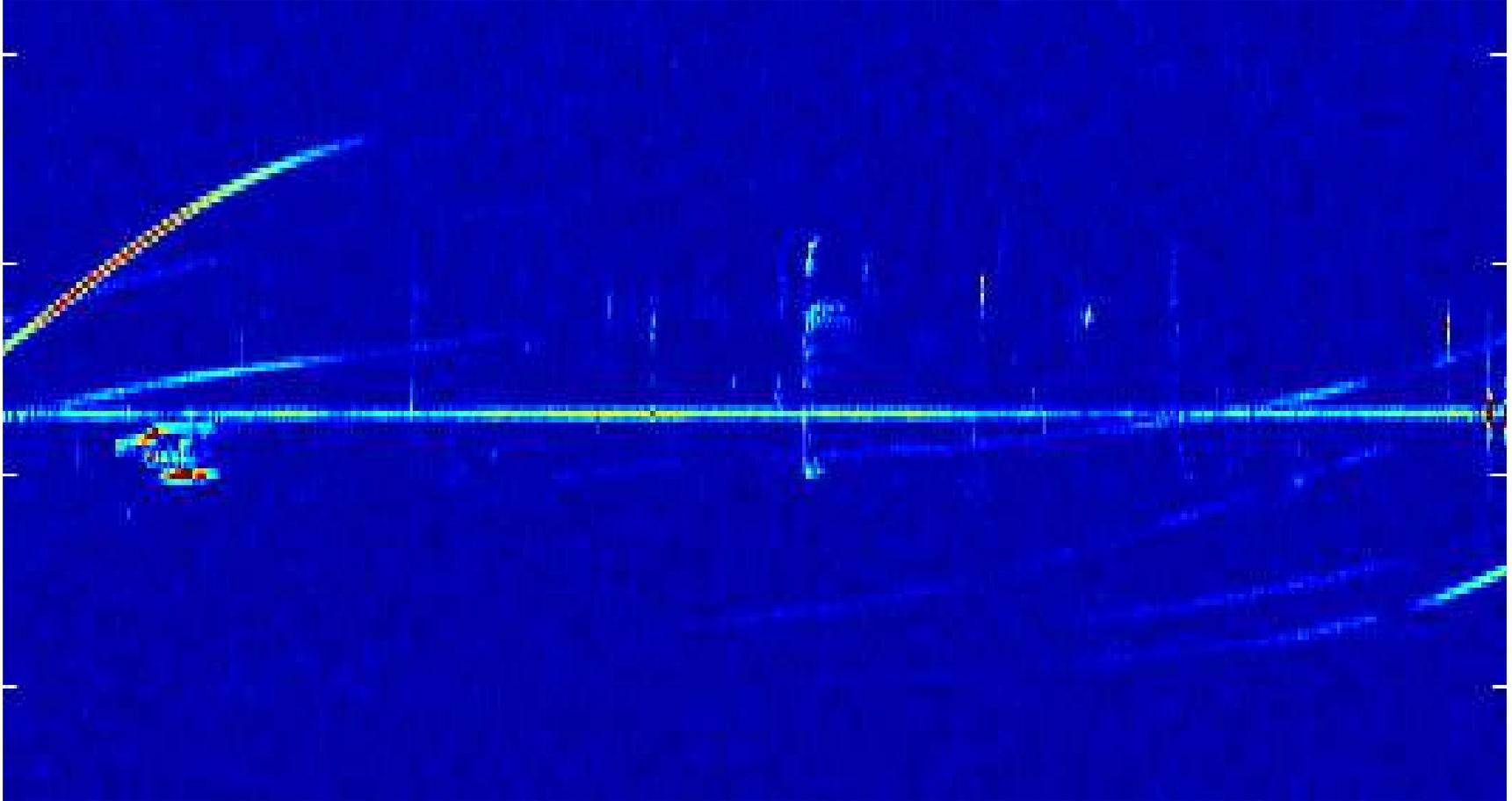


**Essai de détection  
automatique d'échos de  
météores au moyen de  
techniques de traitement  
d'images.**

# Introduction

- Détecter automatiquement un ensemble d'enregistrements d'échos de météores situés dans un répertoire commun
- Sauvegarder le résultat sous forme d'un fichier texte
- A titre tout à fait expérimental
- Fait suite à un Tfe présenté par Mathieu Deltour en juin 2011 à l'EPHEC.  
Utilise l'essentiel de son interface graphique mais modifie sa méthode de détection
- Devrait être réécrit en C ou C++: vitesse d'exécution et portabilité

# Spectrogramme



# Lire le signal

Exemple:

```
[s, Fs]=wavread( ...
```

```
'RAD_BEDOUR_20111008_1935_BEOTTI_SYS001.wav' )
```

s est le signal

Fs est sa fréquence d'échantillonnage qui est de 5512 éch/s pour nos enregistrements.

Le nom du fichier wav doit être du type:

```
xxx_YYYYMMDD_hhmm_xxx.wav
```



```
[S,f,t] = spectrogram(s,window,noverlap,...  
NFFT,Fs,'yaxis');
```

- s est le signal provenant du fichier wav
- f est le vecteur fréquence de Nf lignes
- t est le vecteur temps de Nt colonnes
- S est une matrice de Nf lignes et Nt colonnes
- window est la taille de la fenêtre utilisée pour le calcul du spectrogramme.
- noverlap est un nombre d'échantillons correspondant au recouvrement des fenêtrés lors du calcul du spectrogramme.

Nous prendrons sa valeur par défaut qui est de 50%.

- NFFT est le nombre d'échantillons à partir desquels la FFT (Fast Fourier Transform) est calculée. Nous prendrons sa valeur par défaut qui est égale à la taille de la fenêtre.
- Fs est la fréquence d'échantillonnage
- 'yaxis' permet d'avoir le temps comme axe horizontal

Le spectrogramme sera donc calculé de la manière suivante.

```
[S,f,t] = spectrogram(s,8192,[],[],Fs,'yaxis');
```

Le spectrogramme sera ensuite limité à une bande de fréquence de 300Hz centré essentiellement sur la porteuse.

Le spectrogramme = matrice (donc une image)

- 446 lignes représentant la fréquence
- 400 colonnes représentant le temps

Comme l'enregistrement dure 5 minutes (approximativement) et la plage de fréquence déterminée par le programme est de 300Hz, la valeur d'un pixel sera la suivante:

1.  $\Delta t \simeq \frac{60 * 5}{400} = 0.75s$
2.  $\Delta f \simeq \frac{300}{446} = 0.67Hz$

$\Delta t$  peut être ramené à 0.16s avec un noverlap de 90%

$$\text{Nbre de colonnes} \simeq \frac{\text{tailleSignal}}{\text{tailleFenetre} - \text{noverlap}}$$

Pour Noverlap = 50%:

$$\text{Nbre de colonnes} \simeq \frac{5512 * 300}{8192 - 4096} \simeq 400$$

400 pixels correspondent à 300s d'où 1pixel = 0.75s

Pour Noverlap = 90%:

$$\text{Nbre de colonnes} \simeq \frac{5512 * 300}{8192 - 7323} \simeq 1900$$

1900 pixels correspondent à 300s d'où 1pixel = 0.16s

Notre programme détecte le nombre d'échos de météores underdensés et overdensés enregistrés dans un répertoire donné.

Il permet de sauvegarder le résultat dans un fichier texte donnant le nombre d'échos pour chaque enregistrement.

- Première colonne donne le nom de l'enregistrement
- Deuxième colonne donne le temps en jours depuis l'enregistrement initial
- Troisième le nombre d'échos underdensés et overdensés
- Quatrième colonne fournit le temps initial en jour avec le 01/01/0000 représenté par le nombre 1

201110081600 0.000000000 5 734784.66666667

201110081605 0.00347222 5

201110081610 0.00694444 5

201110081615 0.01041667 7

201110081620 0.01388889 8

201110081625 0.01736111 3

201110081630 0.02083333 7

201110081635 0.02430556 5

# Réduction du bruit

Mettre les valeurs les plus faibles du spectrogramme à 0.  
L'algorithme est le suivant:

```
level_bruit=mean(mean(S));  
[a,b]=size(S);  
for ii=1:a  
    for jj=1:b  
        if (S(ii,jj)<2*level_bruit)  
            S(ii,jj)=0;  
        end  
    end  
end  
end
```

Comme les boucles "for" sont peu efficaces dans Matlab, nous écrirons ceci de la manière suivante:

```
level_bruit=mean(mean(S));  
Indice=find(S<2*level_bruit);  
S(Indice)=0;
```

# Erosion

Pour éroder une image binaire il faut tout d'abord définir un élément structurant.

Celui-ci permet de définir un voisinage pour chacun des pixels de l'image d'origine en leur appliquant l'élément structurant.

Pour une image binaire, si l'un des pixels du voisinage de l'élément structurant est à 0, la valeur de sortie du pixel est alors automatiquement 0.

# Exemple

Soit la matrice M:

M =

1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Soit un élément structurant formé d'un vecteur ligne de 3 éléments de valeur 1:

Il est donné par le code Matlab suivant:

```
>> se=strel('line',3,0)
```

```
se =
```

```
    1    1    1
```

L'érosion de la matrice M par l'élément structurant se est donné par le code Matlab suivant:

```
>> imerode(M,se)
```

element  
structurant

1 1 1

0 0 0 sortie

M =

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1 <sup>0</sup>	0 <sup>0</sup>	1 <sup>0</sup>	0	1	0	0	0
0	0	0	1	0	0	0 <sup>1</sup>	0 <sup>1</sup>	0 <sup>1</sup>	0
1	1	1	1	1	1 <sup>1</sup>	1 <sup>1</sup>	1 <sup>1</sup>	1	1
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

## D'où la solution

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Soit un élément structurant formé d'un vecteur colonne de 3 éléments de valeur 1:

Il est donné par le code Matlab suivant:

```
>> se=strel('line',3,90)
```

```
se =
```

```
1  
1  
1
```

element  
structurant

**M** =

1
1
1

0  
0  
0

sortie

1	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1

## D'où la solution

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Une érosion suivant une ligne verticale de 11 éléments permettra d'éliminer la porteuse et les avions

# Dilatation

Pour une image binaire, si l'un des pixels du voisinage est à 1, la valeur de sortie du pixel est 1.

Soit la matrice:

1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0

Soit un élément structurant formé d'un vecteur colonne de 3 éléments de valeur 1.

Une dilatation appliquée à cette matrice donnera comme résultat:

```
>> imdilate(M1, se)
```

```
ans =
```

```
1 1 0 0 0 0 1 0
1 1 1 0 0 0 1 0
0 1 1 1 0 0 1 0
0 0 1 1 1 0 1 0
0 0 0 1 1 1 1 0
0 0 0 0 1 1 1 0
0 0 0 0 0 1 1 1
0 0 0 0 0 0 1 1
```

# Transformation en image binaire

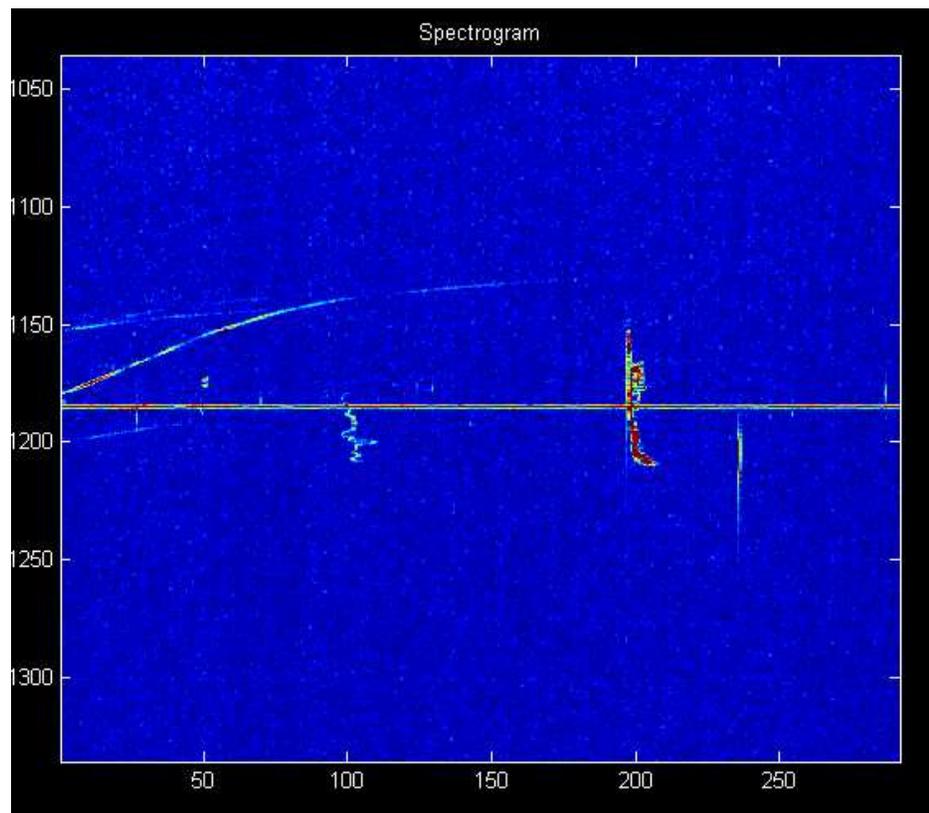
Le niveau est obtenu de manière statistique par l'instruction  
`level=graythresh(S) ;`

La transformation en image binaire est obtenue par  
l'instruction:

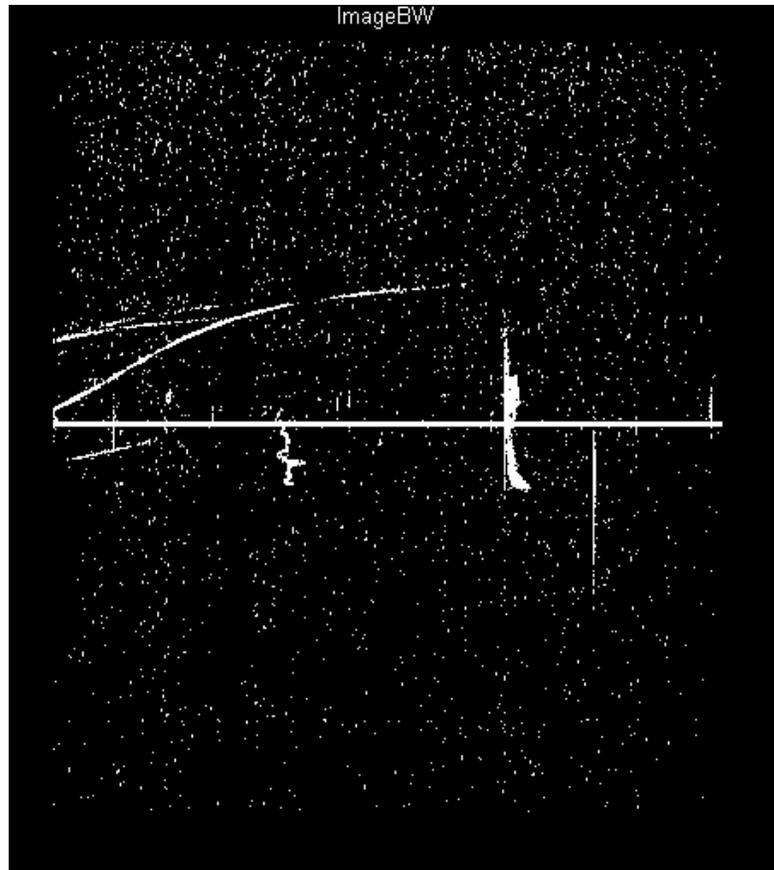
`Sbw=im2bw(S, level) ;`

'RAD\_BEDOUR\_20111008\_1655\_BEOTTI\_SYS001.wav'.

Son spectrogramme est:



Après atténuation du bruit et transformation en image binaire nous obtiendrons:



Après érosion suivant une ligne verticale de 11 éléments, l'image suivante.

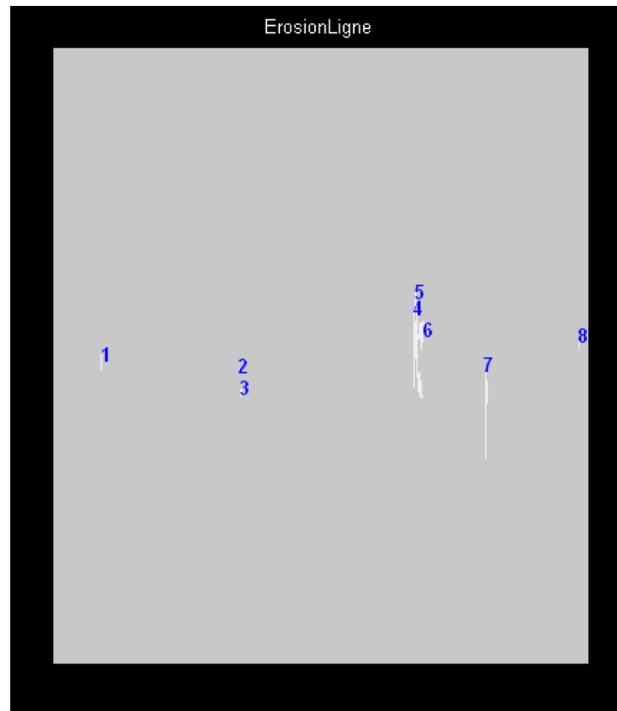


# Labellisation

Nous allons diviser l'image obtenue en objets. Nous en comptons 8.

Nous pourrions compter ceux-ci et leur donner différentes propriétés comme une surface, un centre géométrique, une excentricité...

Nous obtenons 8 objets:



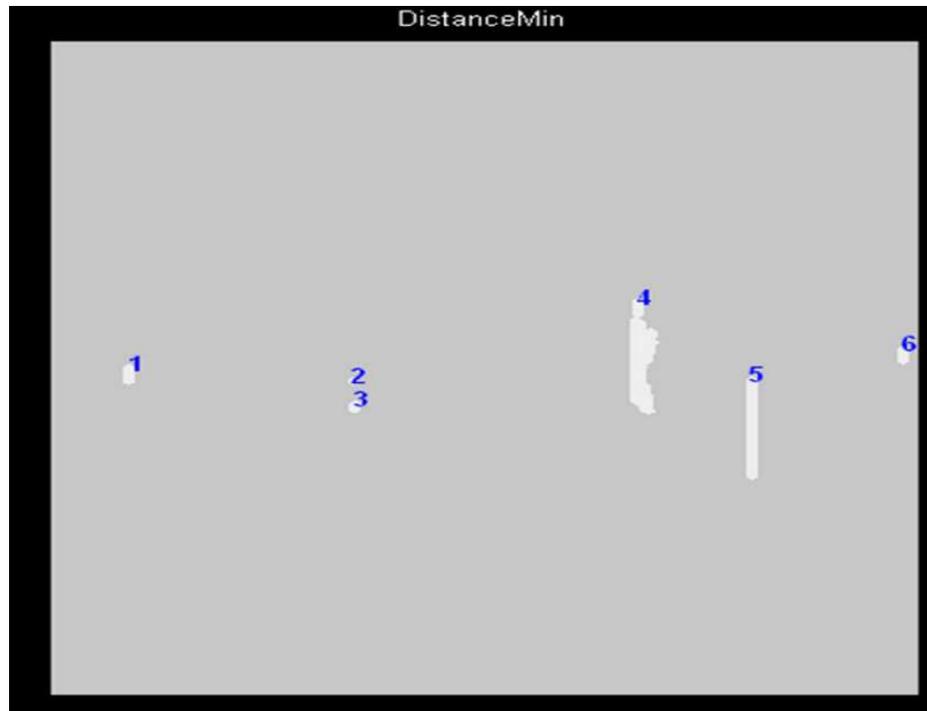
# Deux problèmes à Résoudre

- un objet overdense est compté comme plusieurs underdenses.
- des objets de très faible puissance (de faible valeur du Spectrogramme) sont comptabilisés.

# Eléments “presque connectés”

Deux objets sont presque connectés s’il y a 4 pixels de distance entre eux.

$bw2 = bwdist(Ltotal) \leq 2$  où  $Ltotal$  sont les objets provenant de l’érosion précédente.



# Limitation en puissance

La sélection des objets après atténuation du bruit se fait de manière purement géométrique (érosion).

Elle ne tient pas compte de la valeur du Spectrogramme de ces objets.

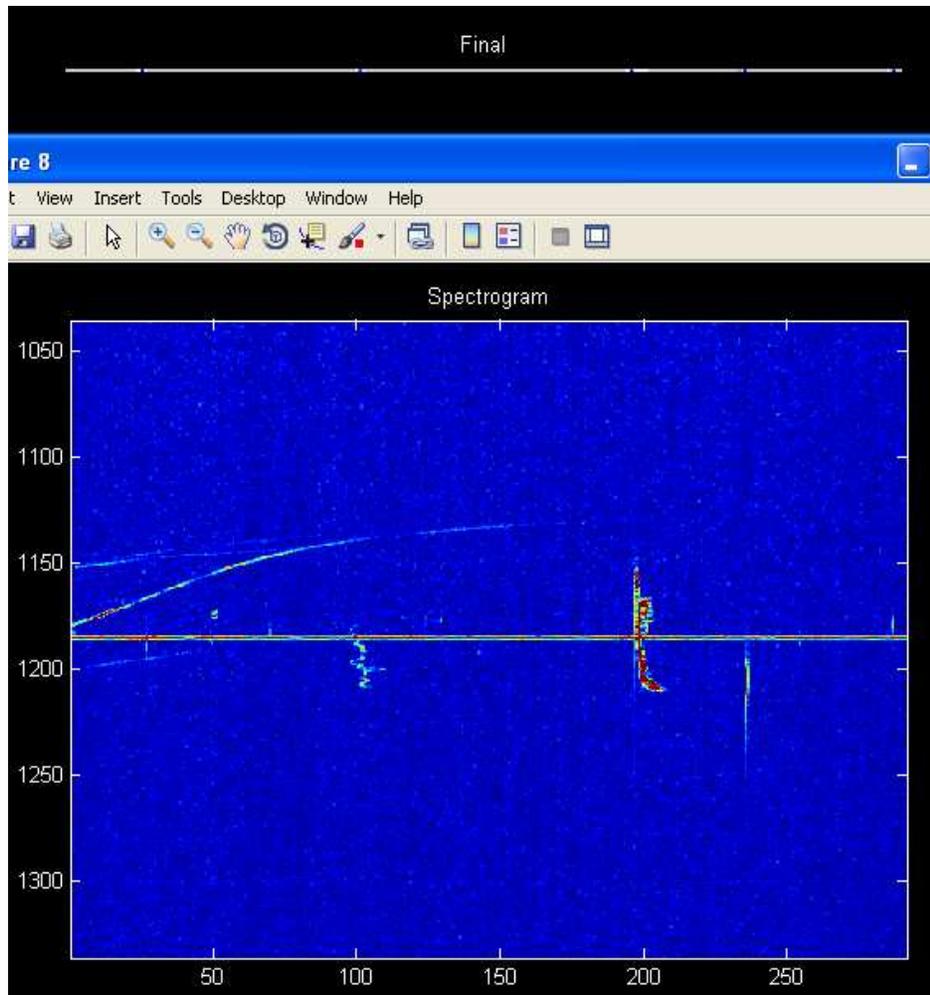
Dans l'exemple précédent, l'écho N°2 est très faible.

Nous allons l'éliminer en ne gardant que les échos pour lesquels le maximum des valeurs du spectrogramme à un temps donné est supérieur à une certaine valeur.

Elle sera choisie, dans notre algorithme comme 5 fois la moyenne du Spectrogramme de départ.

# Résultat final

résultat final: 5 échos dont les valeurs temporelles sont données



NbreCourtLong =

5

temps\_s =

26.8778

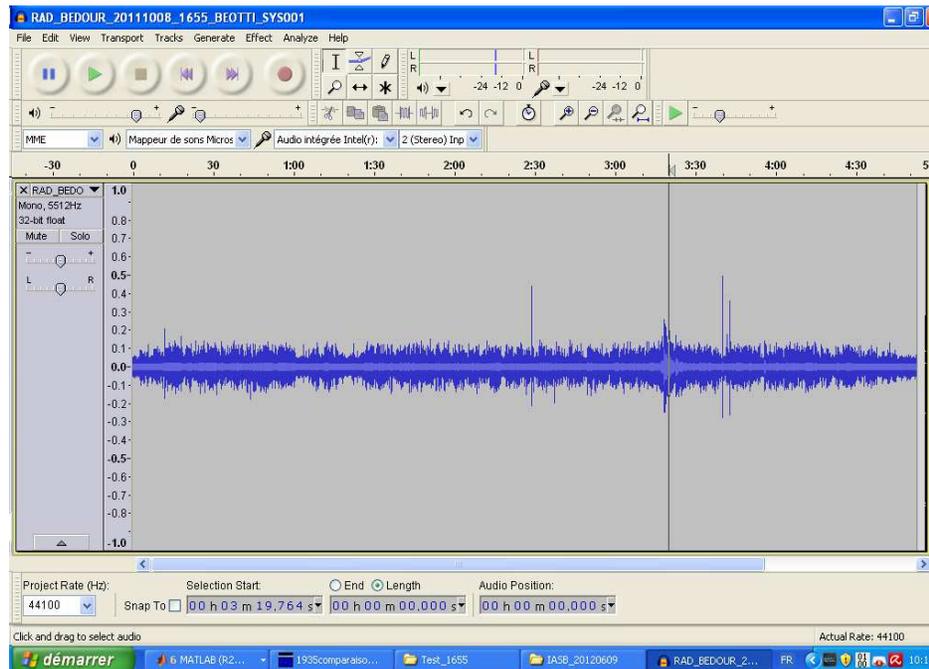
103.4048

200.0901

237.0470

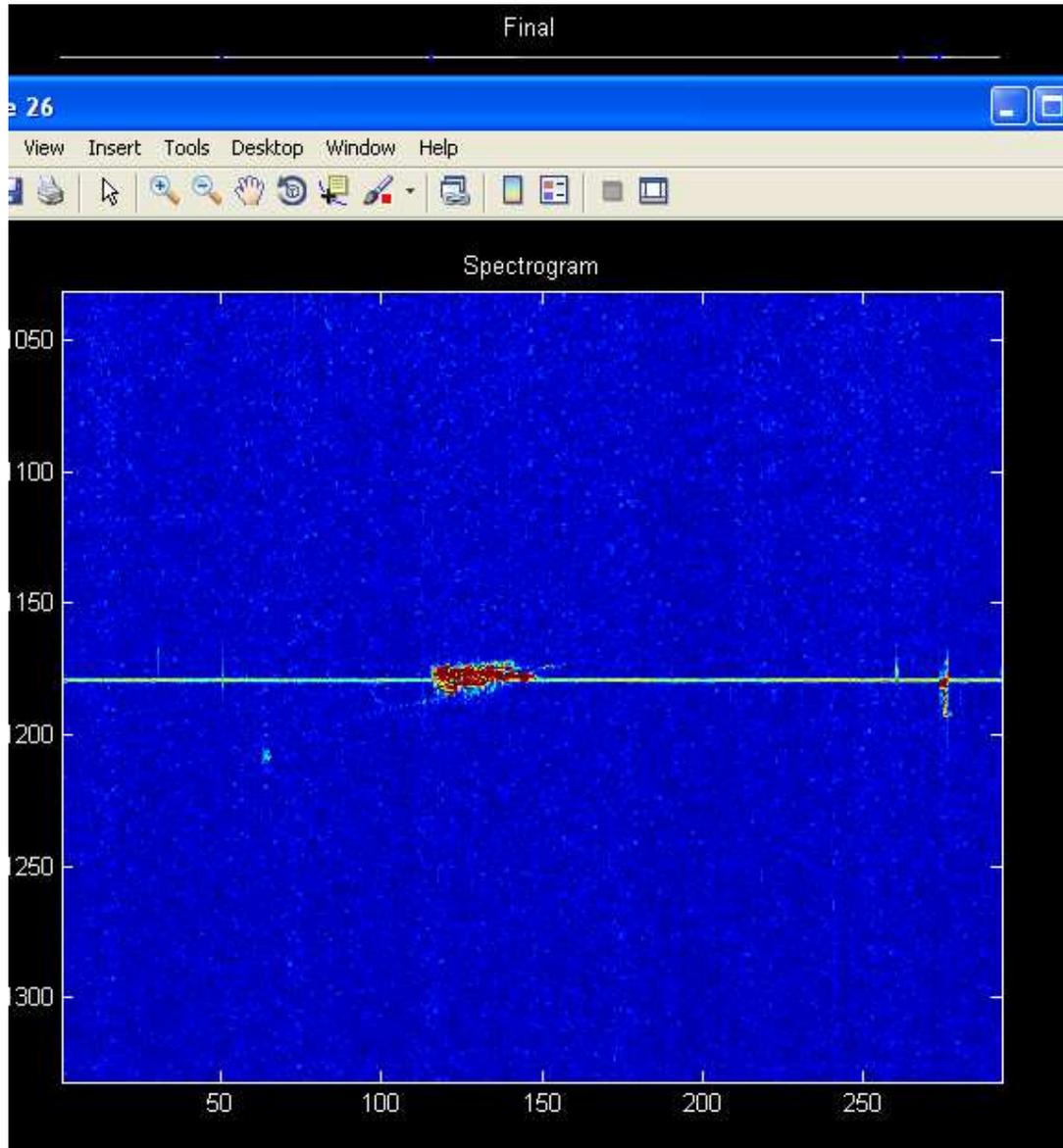
288.1895

Nous pouvons les écouter afin de vérifier notre résultat.



Le premier écho (26.9s) est très faible mais il est audible!

# Autres exemples



NbreCourtLong =

4

temps\_s =

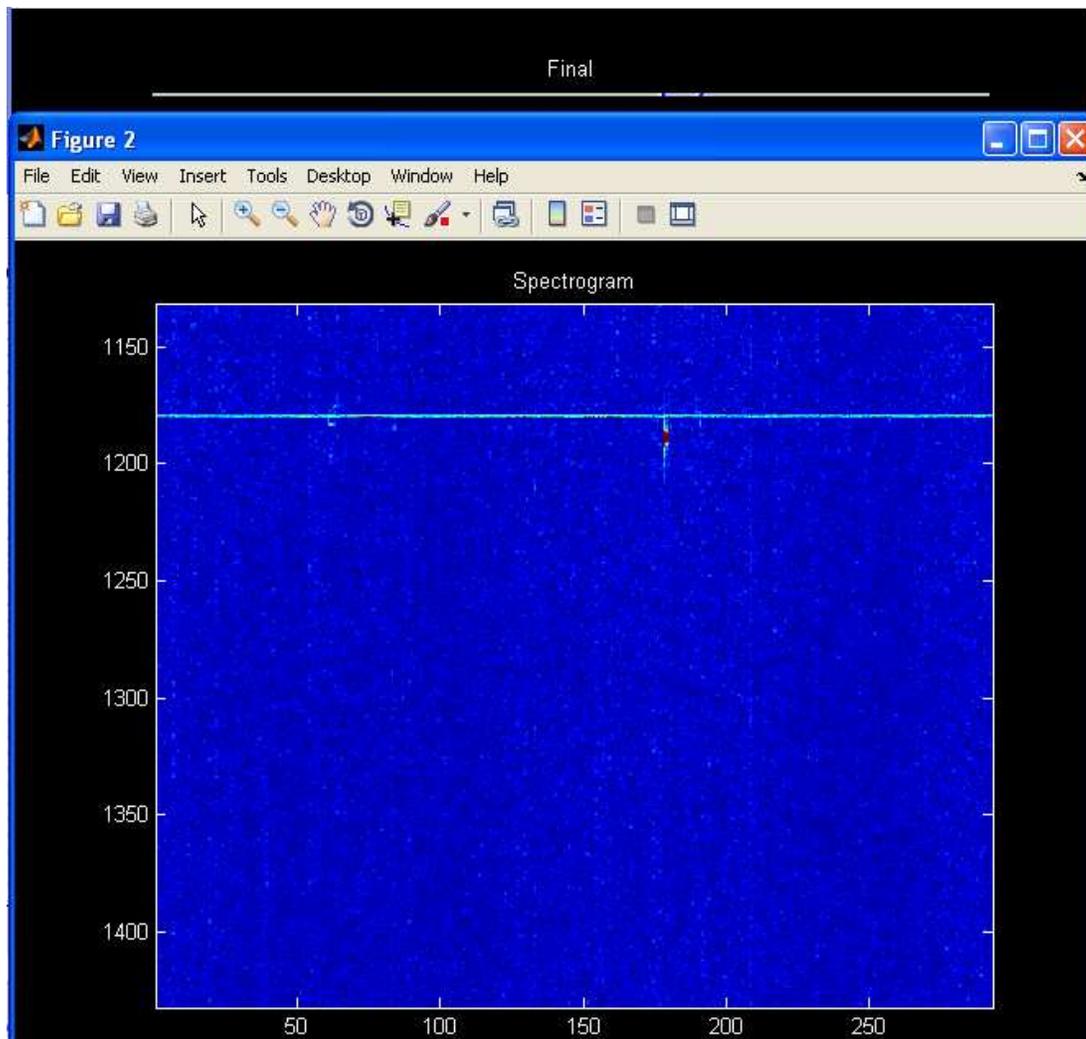
50.7236

129.0468

261.4502

276.3689

RAD\_BEDOUR\_20110801\_0000\_BEOTTI\_SYS001.wav



```
NbreCourtLong =
```

```
2
```

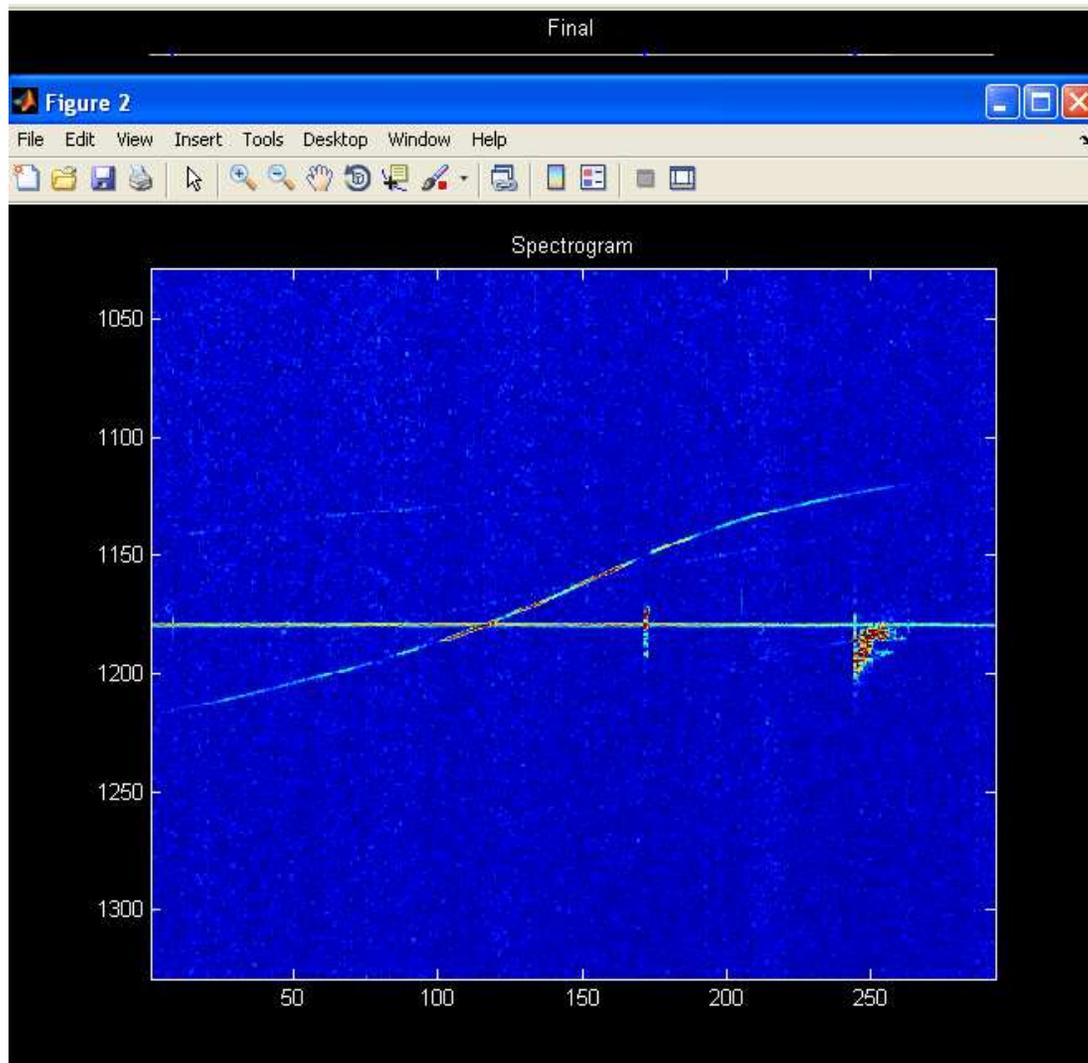
```
temps_s =
```

```
179.0244
```

```
191.7053
```

L'écho de droite très faible mais audible. Echo faible aux alentours de 60s. Est éliminé par l'érosion

RAD\_BEDOUR\_20110801\_0010\_BEOTTI\_SYS001.wav



NbreCourtLong =

3

temps\_s =

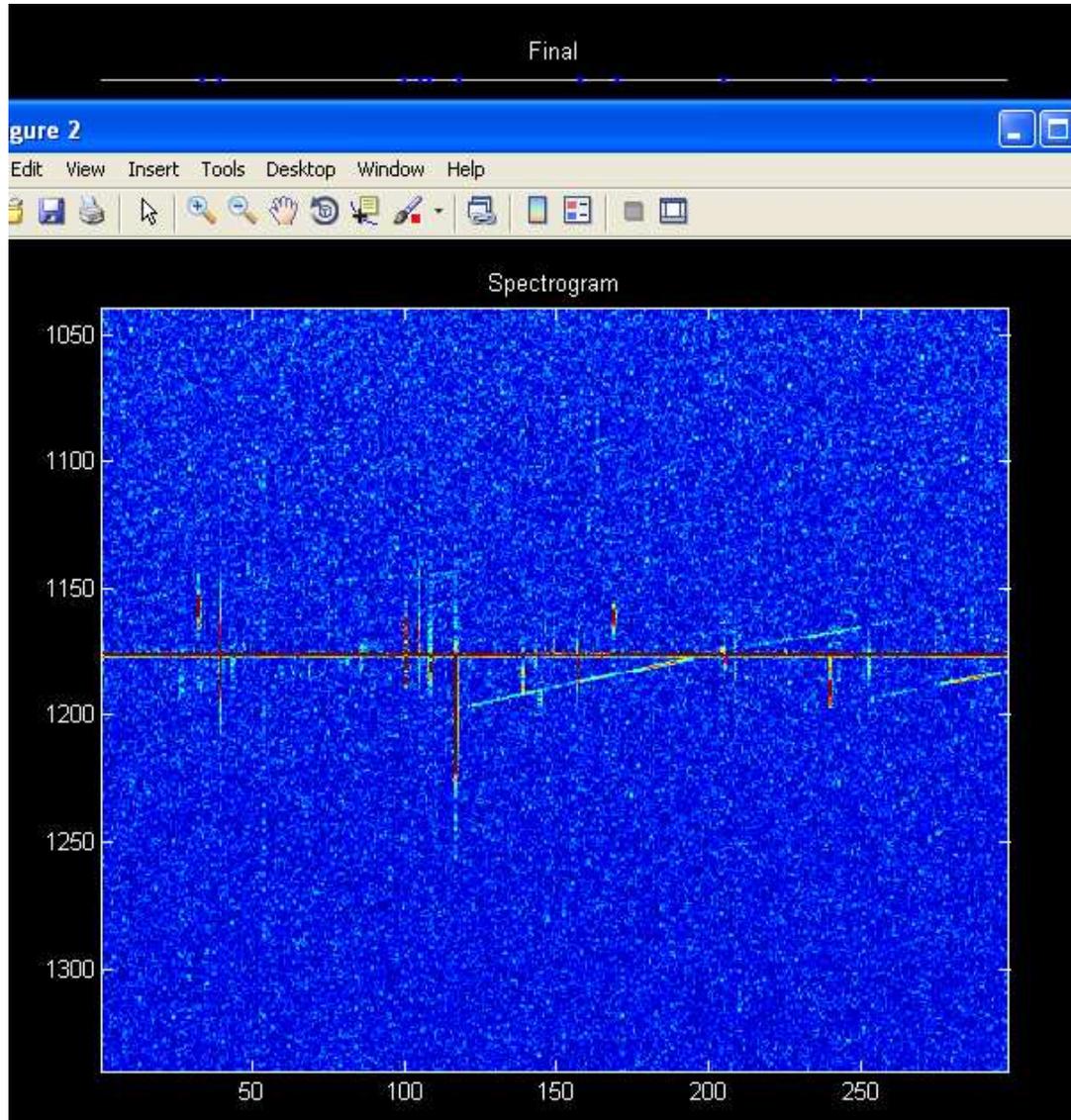
8.2053

172.6840

250.6342

Le premier écho (8s) est très faible mais est audible.

RAD\_BEDOUR\_20120603\_0240\_BEOTTI\_SYS001.wav



NbreCourtLong =

11

temps\_s =

32.0728

39.5316

100.6936

105.1689

108.8983

117.4759

157.3804

169.3145

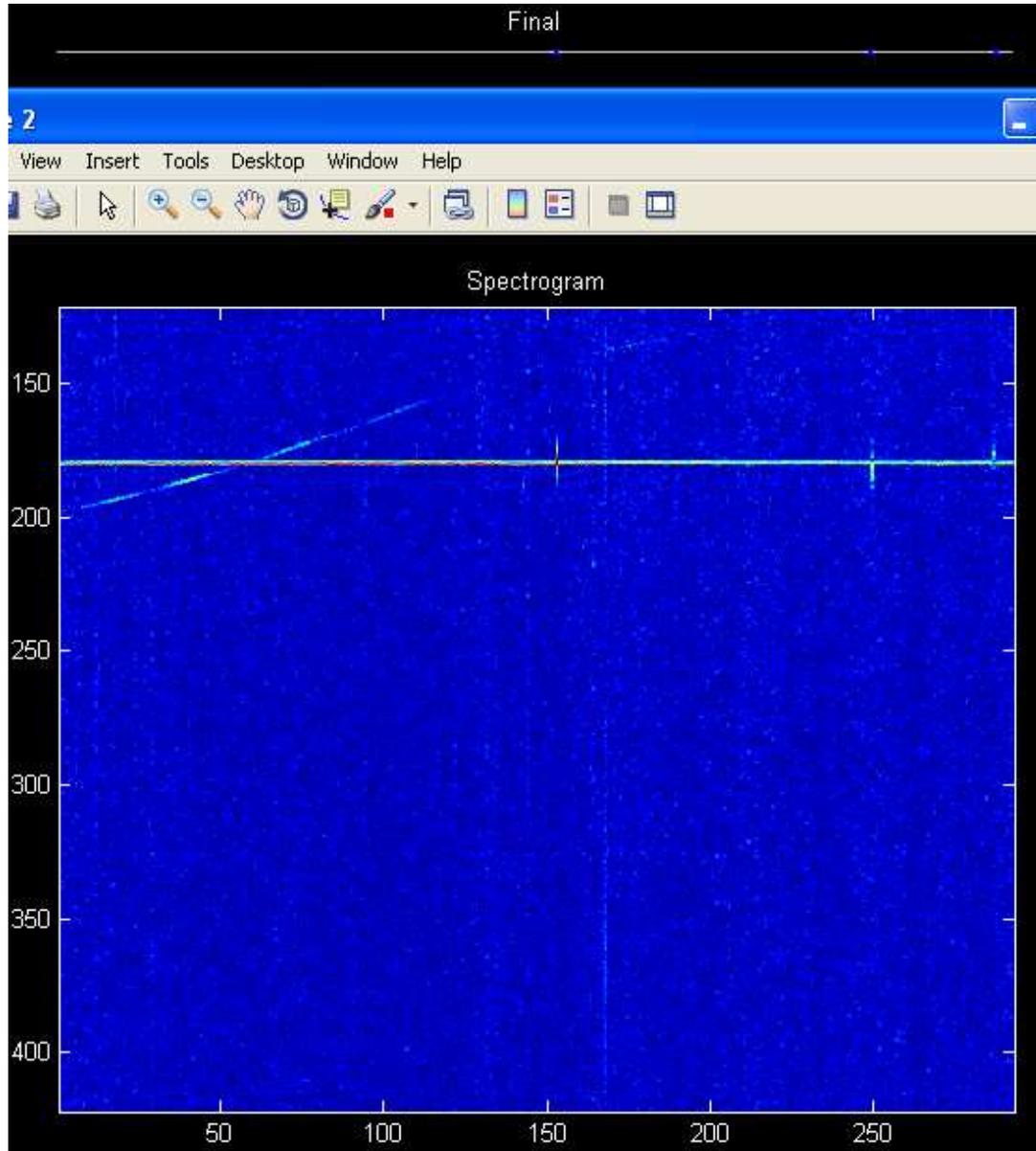
205.8625

240.9188

253.5988

# Parasites

RAD\_BEDOUR\_20110801\_0020\_BEOTTI\_SYS001.wav



NbreCourtLong =

3

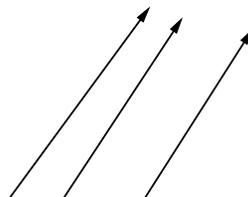
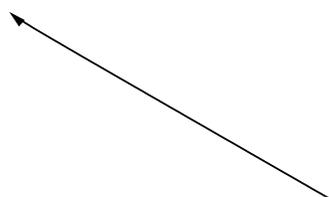
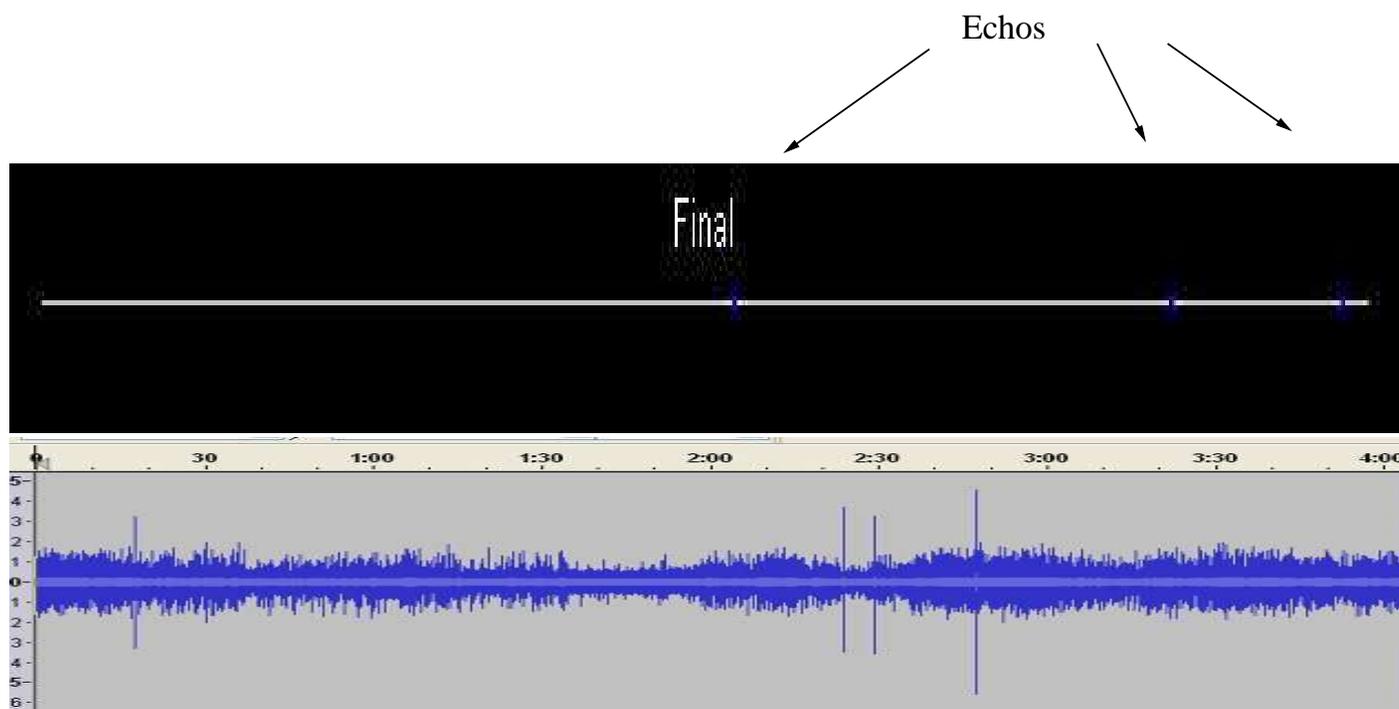
temps\_s =

153.6626

250.6342

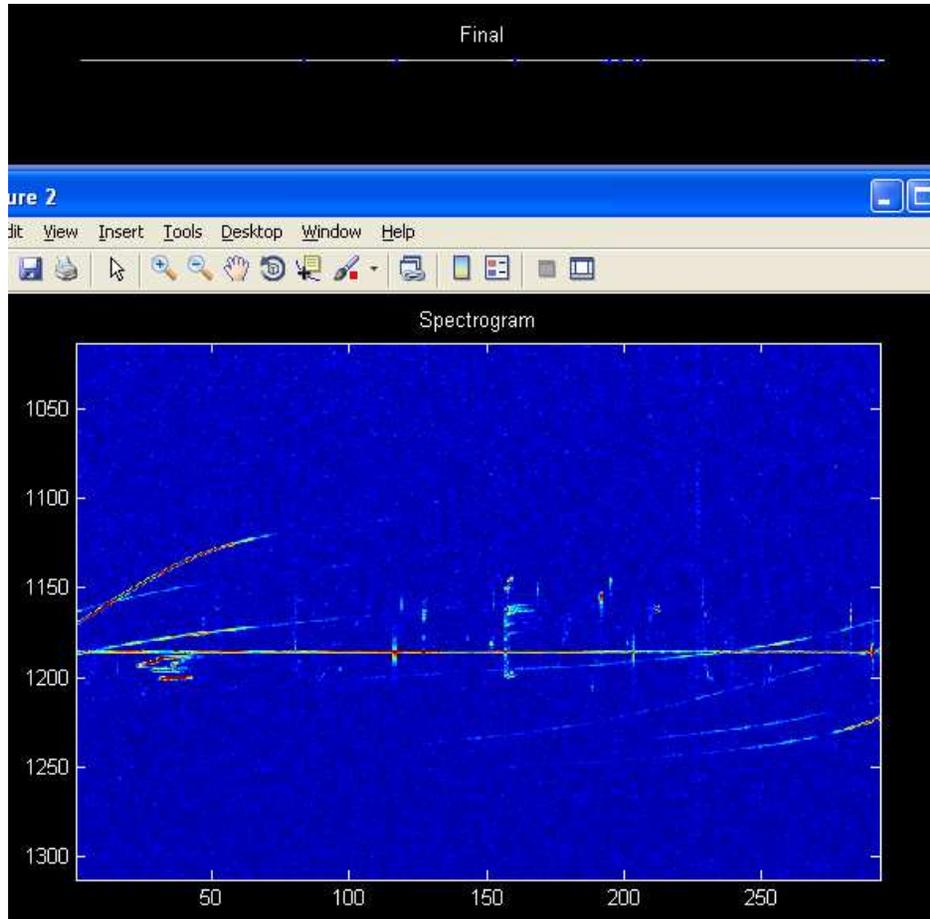
287.5580

Dans cet exemple, les parasites ne sont pas détectés.  
Leurs colonnes dans le Spectrogramme ne possèdent pas 11 éléments continus après débruitage.



# Cas litigieux

RAD\_BEDOUR\_20110810\_1935\_BEOTTI\_SYS001.wav



```
MbreCourtLong =
```

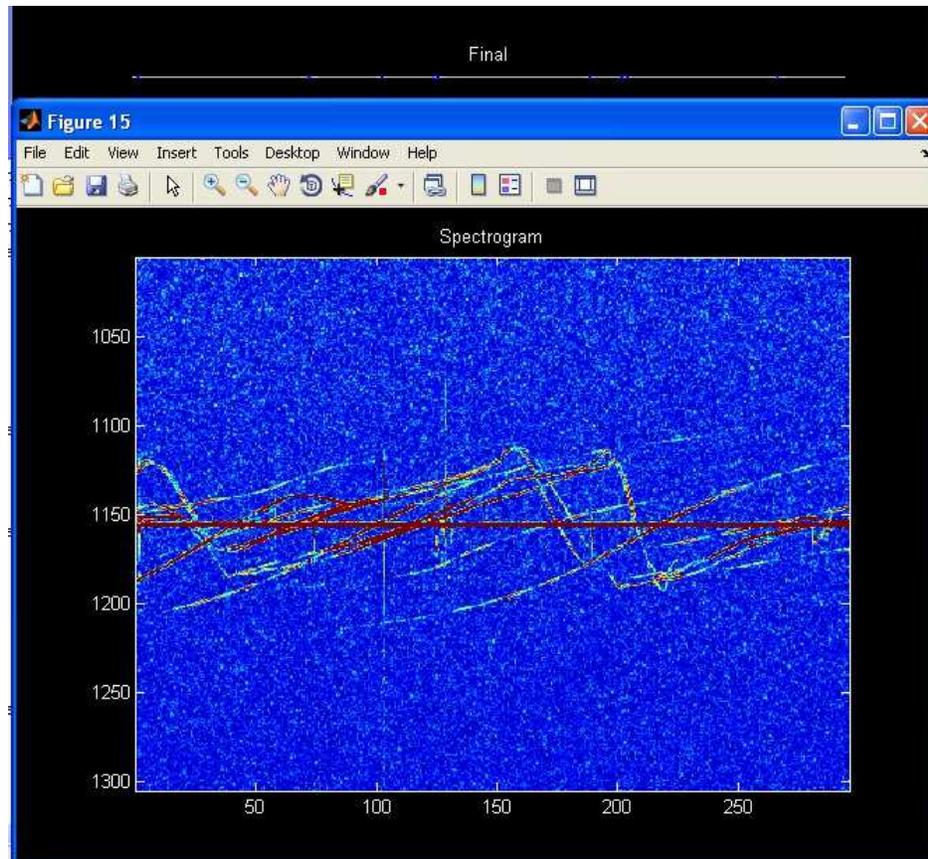
```
8
```

```
temps_s =
```

```
80.5610  
117.1118  
158.1382  
192.0783  
195.8079  
204.3862  
283.4553  
291.2876
```

le premier écho overdense (epsilon), n'est pas détecté. Il ne possède pas une ligne verticale de 11 éléments!

RAD\_BEDOUR\_20120315\_1010\_BEOTTI\_SYS001.wav



```
NbreCourtLong =
```

```
7
```

```
temps_s =
```

```
1.8636
```

```
74.9167
```

```
103.6161
```

```
128.2157
```

```
190.0872
```

```
205.7414
```

```
270.2219
```

Des fichiers possèdent beaucoup d'avions et des intersections d'avions. Certaines d'entre-elles peuvent être détectée comme écho.

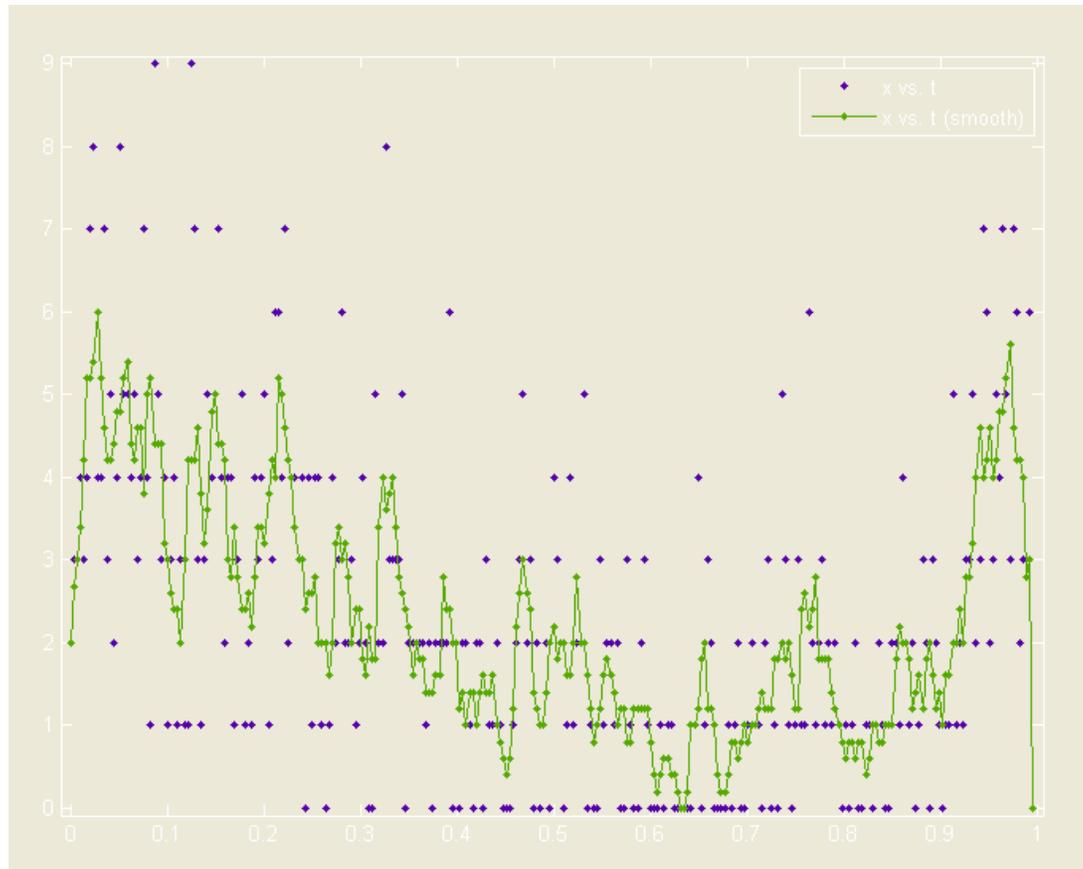
# Statistiques

Le programme permet d'exporter les résultats sous la forme d'un fichier texte.

Des statistiques journalières, par exemple, peuvent être obtenues.

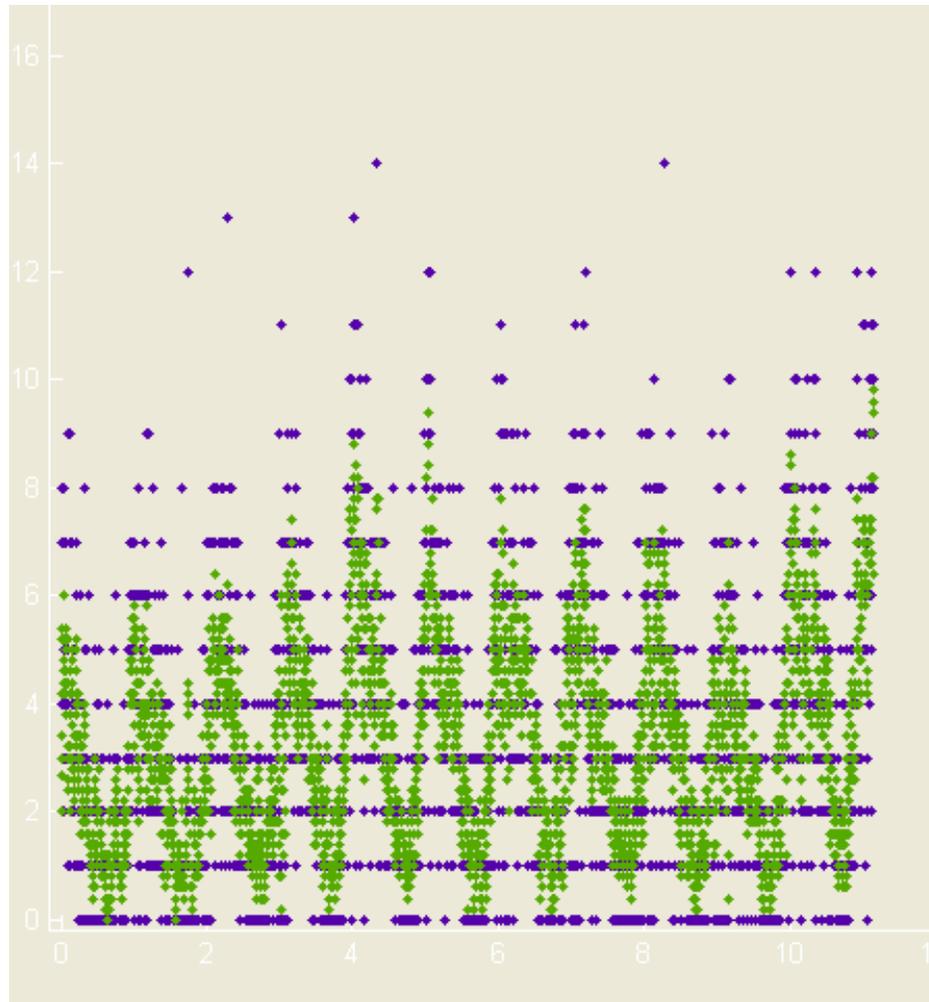
Il faut environ 5 minutes pour calculer les statistiques d'une journée d'enregistrements. La principale difficulté provient de la mauvaise qualité d'enregistrements en milieu de journée à cause de la grande quantité d'avions ou de parasites.

# 20110801



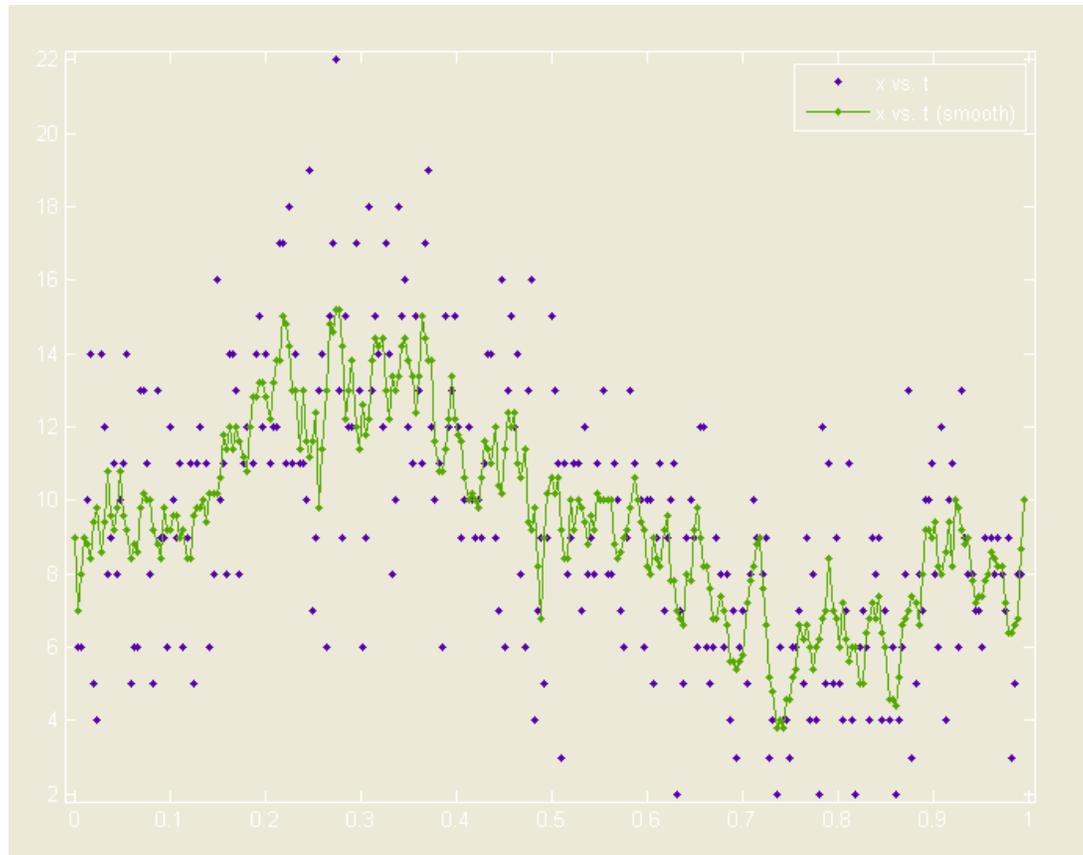
On constate une variation journalière

# 20110801 à 20110810



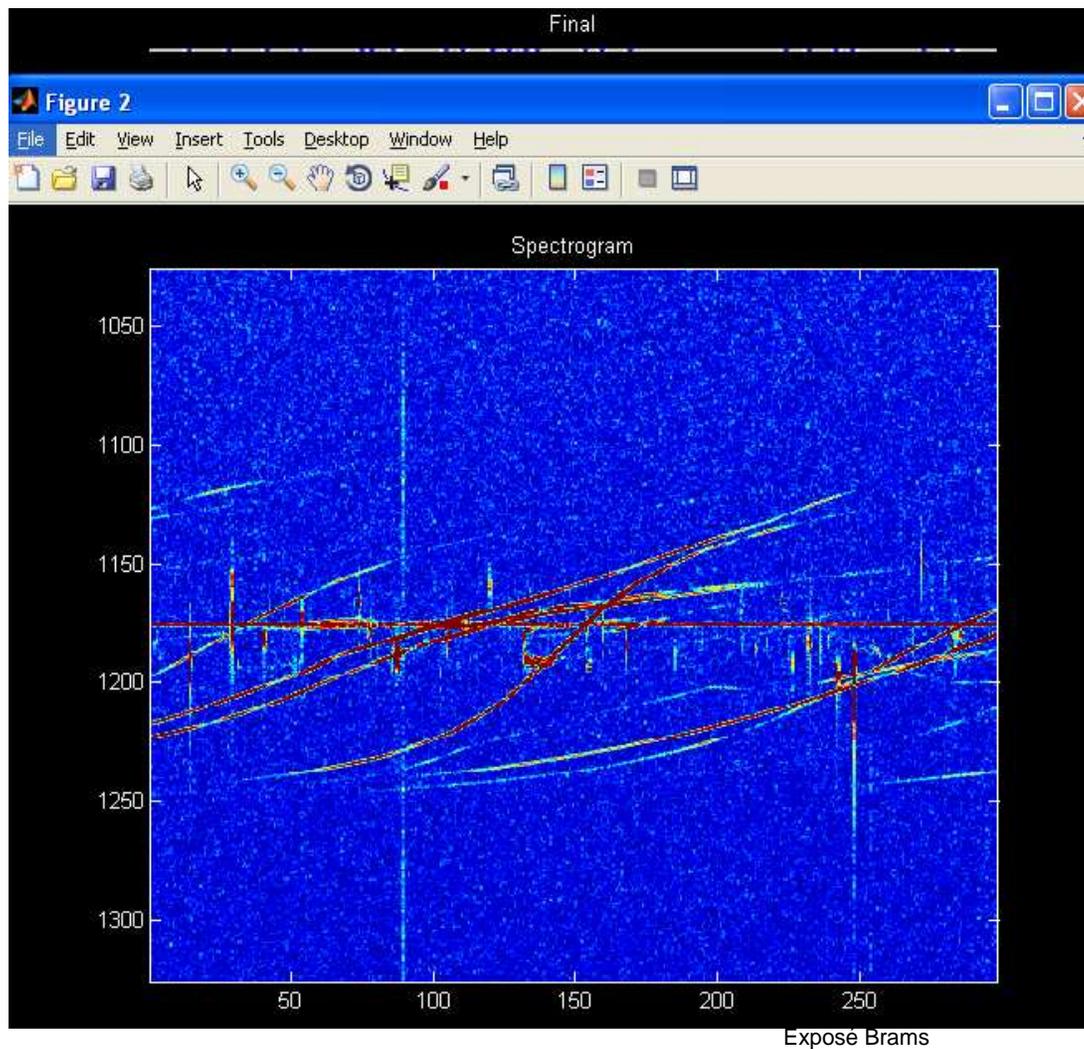
On constate une périodicité journalière.

# 201200603



On constate une grande activité avec un maximum à 6h35:  
les Arietiden?

RAD\_BEDOUR\_20120603\_0635\_BEOTTI\_SYS001.wav



NbreCourtLong =

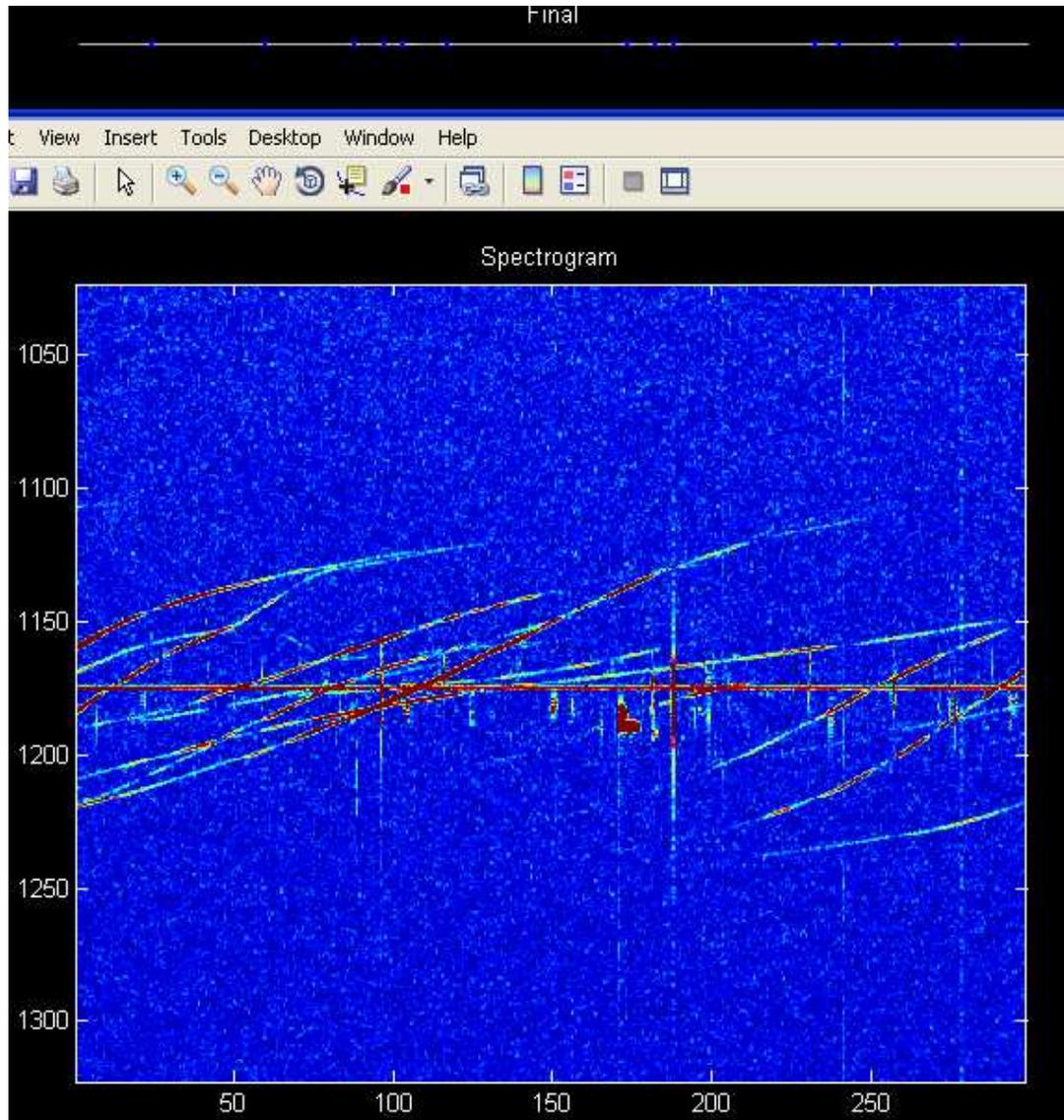
22

temps\_s =

14.1805  
28.3610  
40.6756  
53.7365  
74.2609  
77.9926  
87.3219  
105.2341  
111.5780  
120.1609  
127.6243  
132.4755  
137.3267  
154.4926  
160.4633  
168.6730  
223.9023  
233.6047  
243.6803  
249.6510  
273.1608  
285.1022

09/06/2012

RAD\_BEDOUR\_20120603\_0710\_BEOTTI\_SYS001.wav



NbreCourtLong =

13

temps\_s =

21.6439

58.9609

87.3219

96.6511

103.7414

116.4292

173.8974

182.1072

188.4511

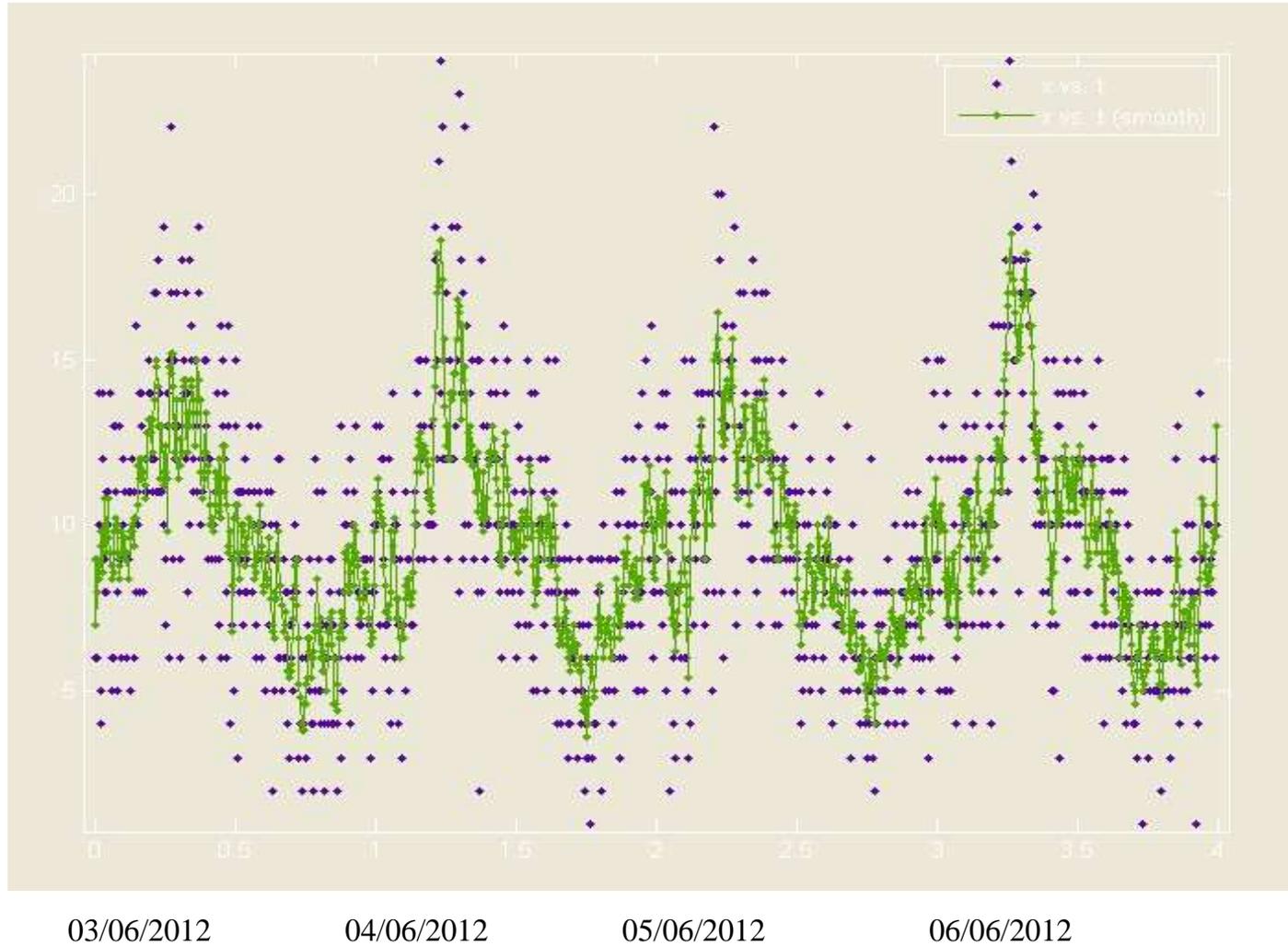
231.7388

238.8291

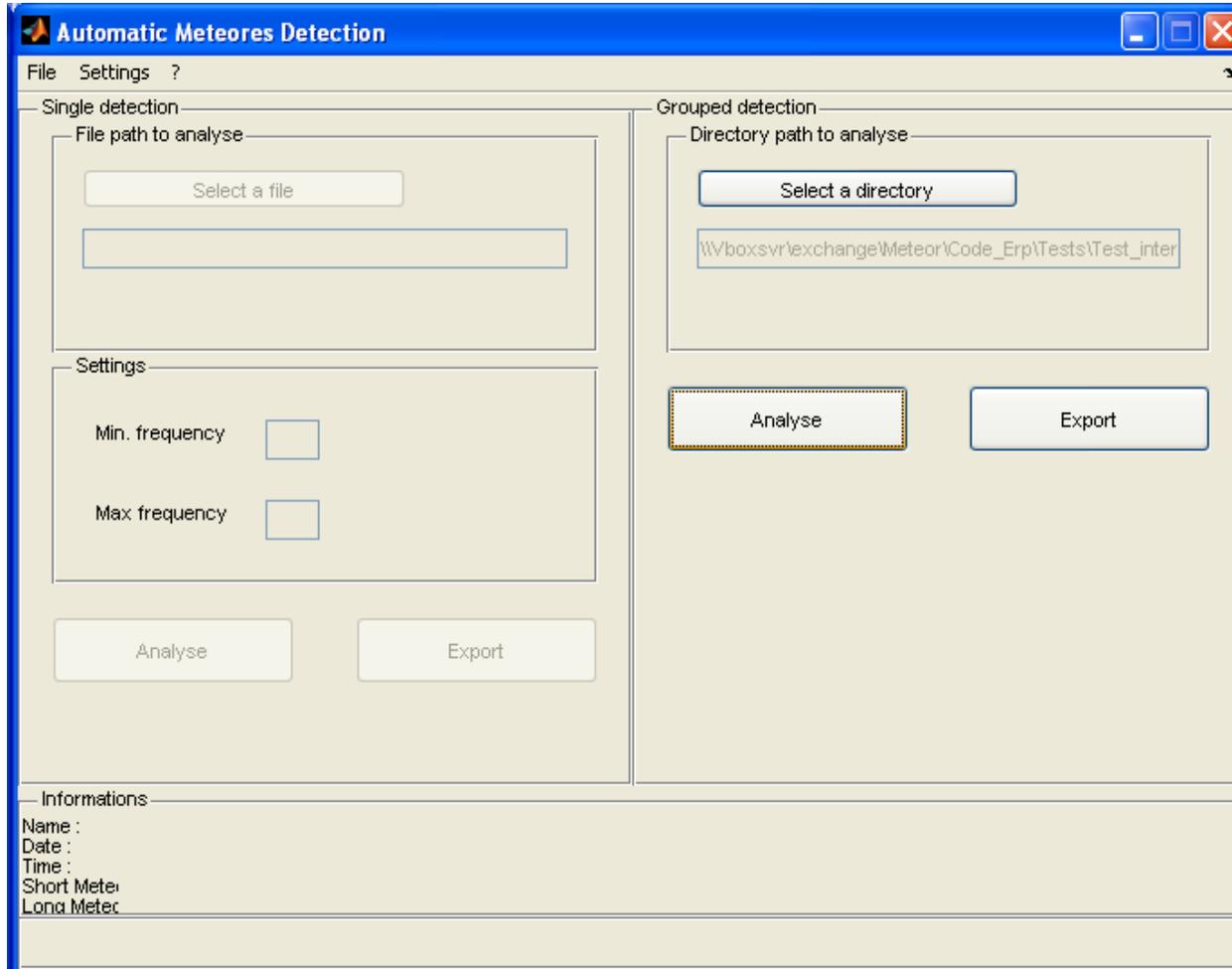
258.2339

277.6388

# 3 au 6 juin 2012



# Le programme



- Il faut placer les fichiers à analyser dans un répertoire, le sélectionner puis demander l'analyse.  
Un bouton permet d'exporter les résultats vers un fichier texte.  
Une version du programme nommé "debug" fournit les graphiques permettant de visualiser les échos détectés.
- Un programme nommé "statistique" élimine tous les graphiques et ne produit que le fichier texte.

# Conclusion

- Notre algorithme est très simple et semble bien détecter la somme des échos underdensés et overdensés.
- Il ne permet pas de détecter les échos overdensés seuls
- Il est peu efficace s'il y a trop d'intersections d'avions
- Il contient un certain nombre de paramètres arbitraires qui devraient être objectivés.  
4 paramètres sont utilisés:

- 8192 pour la dimension de la fenêtre pour les spectrogrammes. Cette valeur est adéquate pour notre fréquence d'échantillonnage de 5512 éch/s
- 50% pour noverlap
- 2 pour la réduction de bruit
- 11 pour l'érosion suivant une ligne verticale
- 2 pour la distance euclidienne
- 5 pour la réduction en puissance