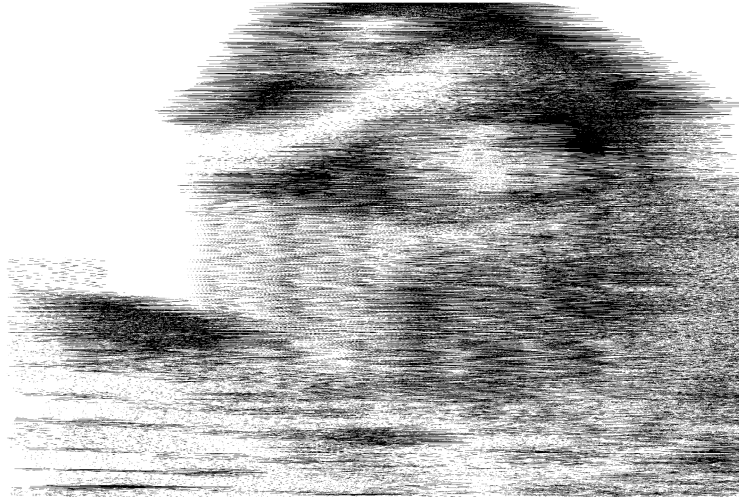


1 *Sujet 1*

On considère une image, dont les lignes ont été décalées de quelques pixels (+/- 500 pixels à droite ou à gauche). L'image est présentée ci-dessous.



Si vous travaillez avec Scilab

Cette image correspond au fichier *image7.mat* (si pas assez de mémoire, *image8.mat*) à récupérer sur le site <http://easytp.cnam.fr/terre>

Vous pouvez le lire et afficher cette image de la manière suivante :

```
clear all;
stacksize('max');
loadmatfile("-mat", 'image7');
Matplot(C);
```

Vous devez corriger les décalages des lignes de cette image. Pensez à la fonction **corr**, exemple :

```
r=corr(B(k,:),B(k-1,:),500);
```

Vous ferez attention à considérer les cas des décalages positifs et négatifs.

Si vous travaillez avec Matlab

Cette image correspond au fichier *fichier2.bmp* à récupérer sur le site <http://easytp.cnam.fr/terre>

```
B=imread('fichier1.bmp','bmp');
B=255*B;
image(B);
colormap(GRAY);
```

2 *Sujet 2*

On vous donne un signal $x(n)$. (à récupérer sur le site <http://easytp.cnam.fr/terre> , fichier *ErlkBruite.wav*)

Ce signal a été bruité par une fréquence mobile. On vous propose de lutter contre cette fréquence au moyen d'un filtre de prédiction linéaire adaptatif.

$$e(n) = x(n) - A(n-1)^T X(n-1)$$

$$A(n) = A(n-1) + \delta e(n) X(n-1)$$

Vous choisirez un prédicteur à deux coefficients et un pas $\delta = 0.1$

3 *Bonus*

Décodez le signal *Encode.wav*