IMAGES NUMERIQUES

Stéphane Verjux – IREM de Franche Comté - 2017

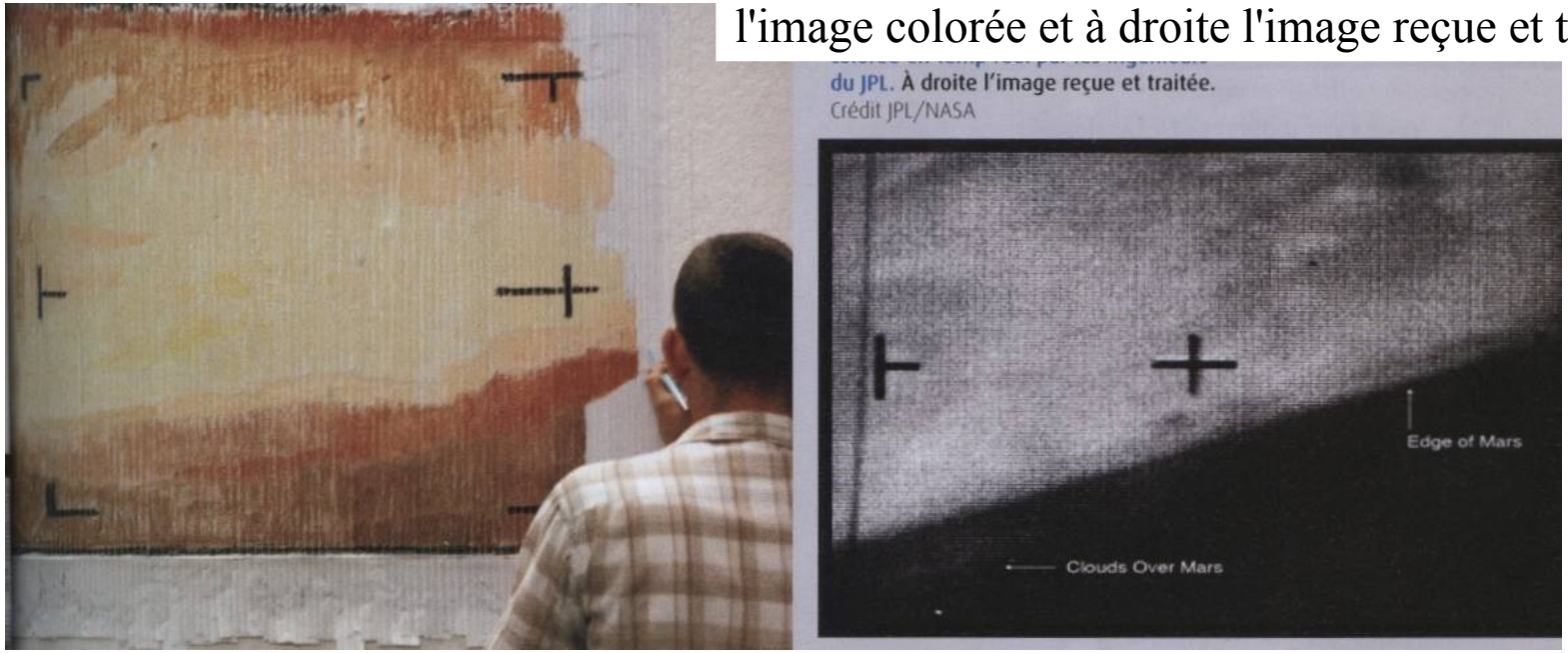
Les éléments d'image ou pixels

Les images numériques sont formées de milliers ou de millions de petits carrés de teinte uniforme appelés **pixels**, contraction de l'expression anglaise ***picture elements***, autrement dit « éléments d'image ».

Ces carrés doivent être assez petits pour que l'on ne puisse pas les distinguer sur une image examinée dans des conditions normales.

les pixels apparaissent lorsque l'on agrandit beaucoup les images. Ils disparaissent si l'on s'éloigne suffisamment de l'écran.

Il existe une anecdote à propos des premières images reçues de Mariner 4. Ne voulant pas attendre le traitement électronique (long) des images reçues, les ingénieurs les ont traitées en temps réel lors de leur arrivée. **Chaque image comprenait 200 lignes de 200 pixels**, chaque point représentant **64 nuances de gris**. À chaque groupe de valeurs de gris, ils ont fait correspondre arbitrairement une couleur sur une bande de papier qui était ensuite collée sur un grand tableau. On aboutissait ainsi à une image en fausses couleurs de la planète. On voit à gauche l'image colorée et à droite l'image reçue et traitée.



L'appareil photographique numérique, l'ordinateur, l'écran ou l'imprimante travaillent donc un peu à la manière des peintres impressionnistes dont les tableaux étaient faits de minuscules touches de peinture, ou encore des mosaïstes qui assemblent des milliers de petits morceaux de pierres ou de céramiques colorées.

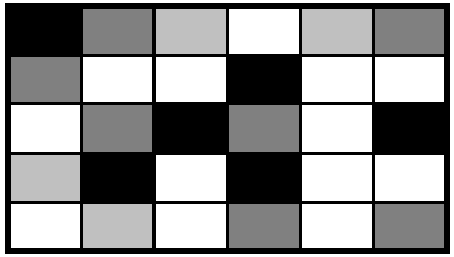
Paul Signac, le Palais des Papes à Avignon

Cheval marin, bains romains de Bath

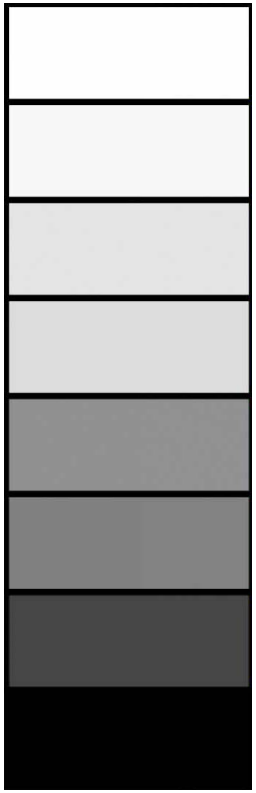

numerisation_1
+
-
Zoom = fois 5
numerisation_1
+
-
Zoom = fois 11
numerisation_1
+
-
Zoom = fois 11
numerisation_1
+
-
Zoom = fois 38

Lecture

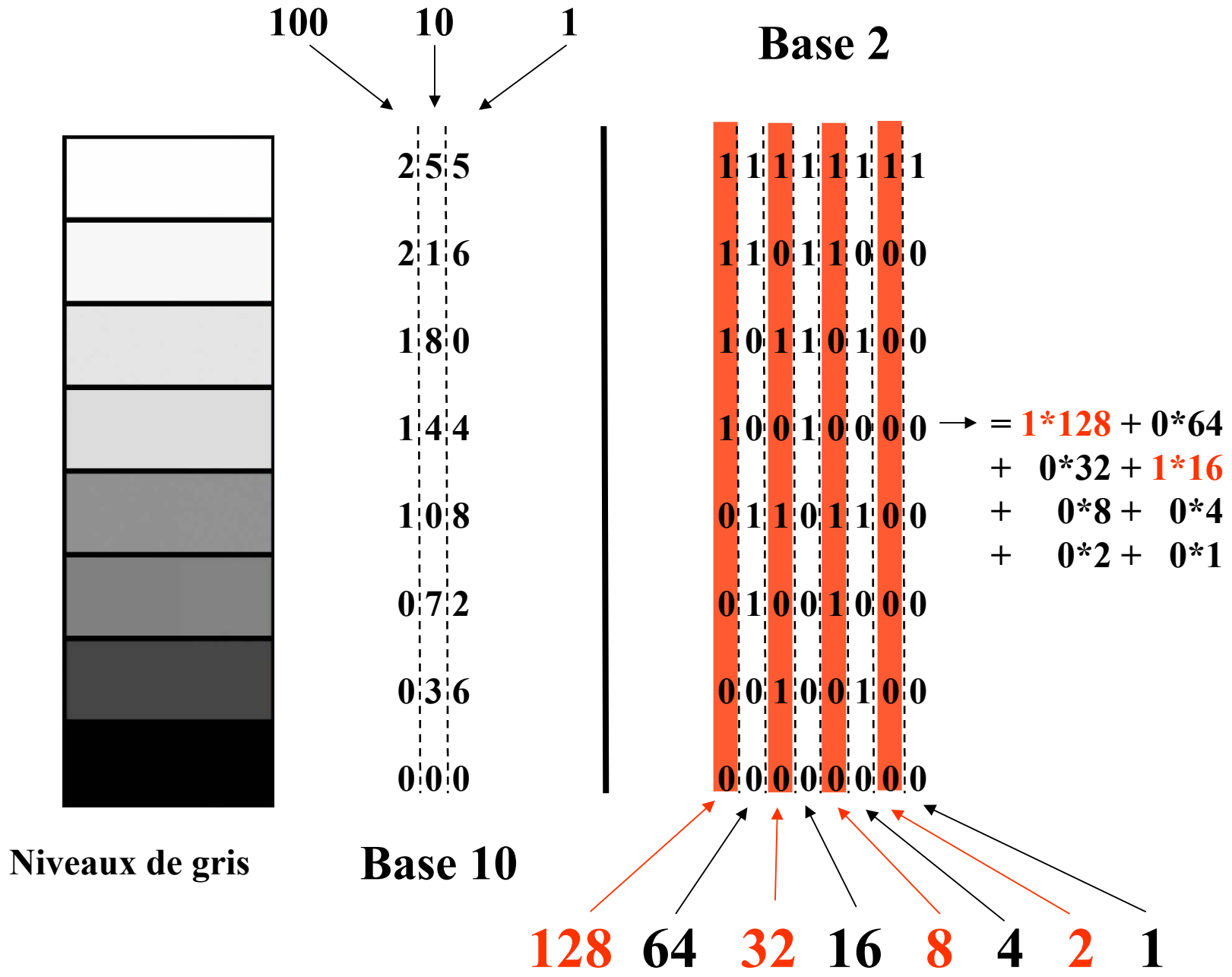
Pour afficher une image numérique, l'ordinateur divise l'écran ou la page de l'imprimante en un réseau de petits carrés et attribue à chacun d'eux, à partir des données mises en mémoire, une luminosité ou une couleur. L'image est ainsi définie point par point et l'on obtient une « **carte de points** » ou « ***bitmap*** » en anglais.



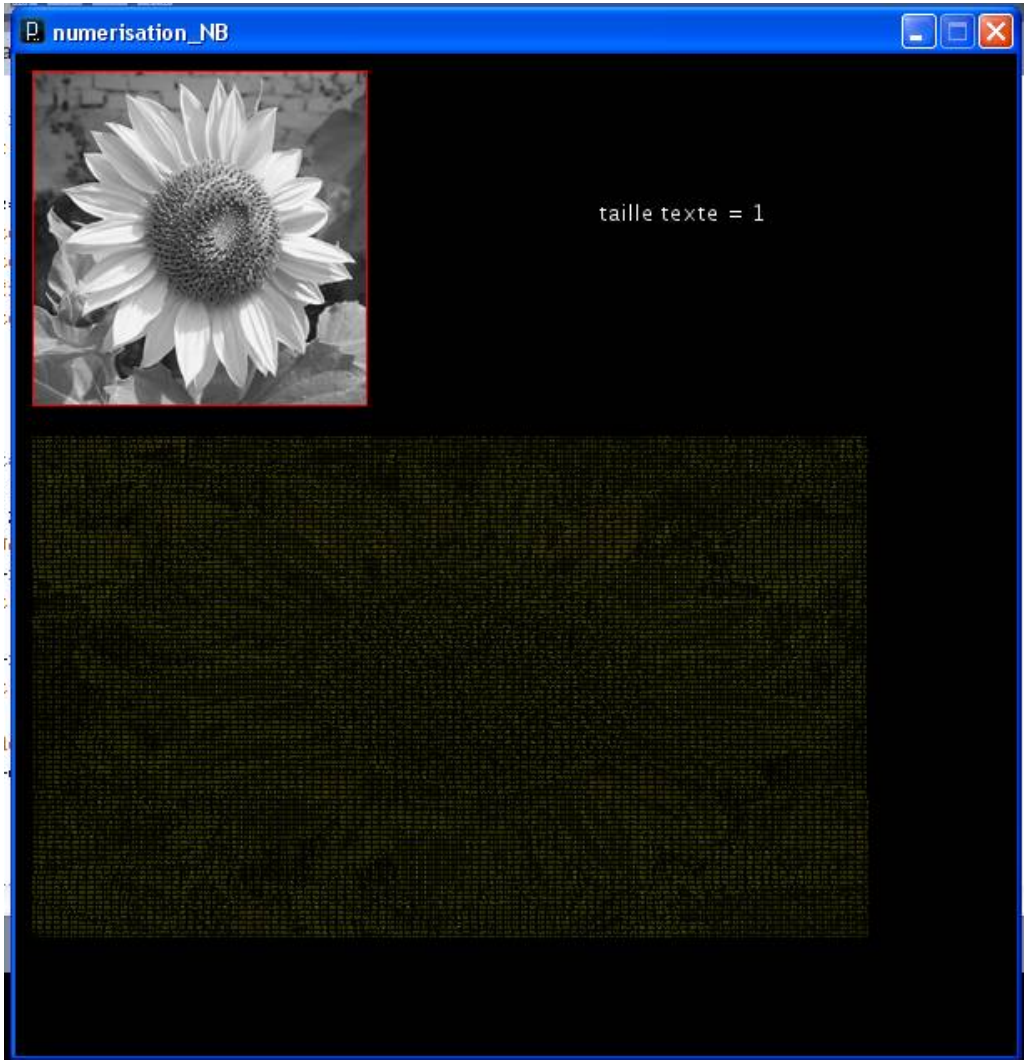
Chaque couleur est ensuite **codée**, numérisée, c'est-à-dire remplacée par un nombre…



Niveaux de gris	Base 10	Base 2
Blanc	255	11111111
	216	11011000
Gris clair	180	10110100
	144	10010000
Gris foncé	108	01101100
	072	01001000
	036	00100100
Noir	000	00000000


100
10
1
Base 2
255
216
180
144
108
072
036
000
11111111
11011000
10110100
10010000
01101100
01001000
00100100
00000000
= 1*128 + 0*64
+ 0*32 + 1*16
+ 0*8 + 0*4
+ 0*2 + 0*1
Niveaux de gris
Base 10
128 64 32 16 8 4 2 1

Une image numérique = un tableau de nombres

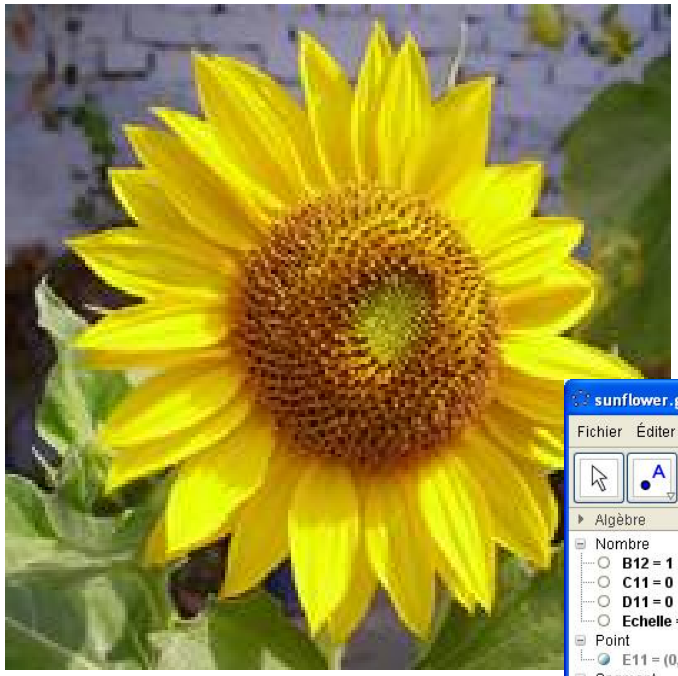


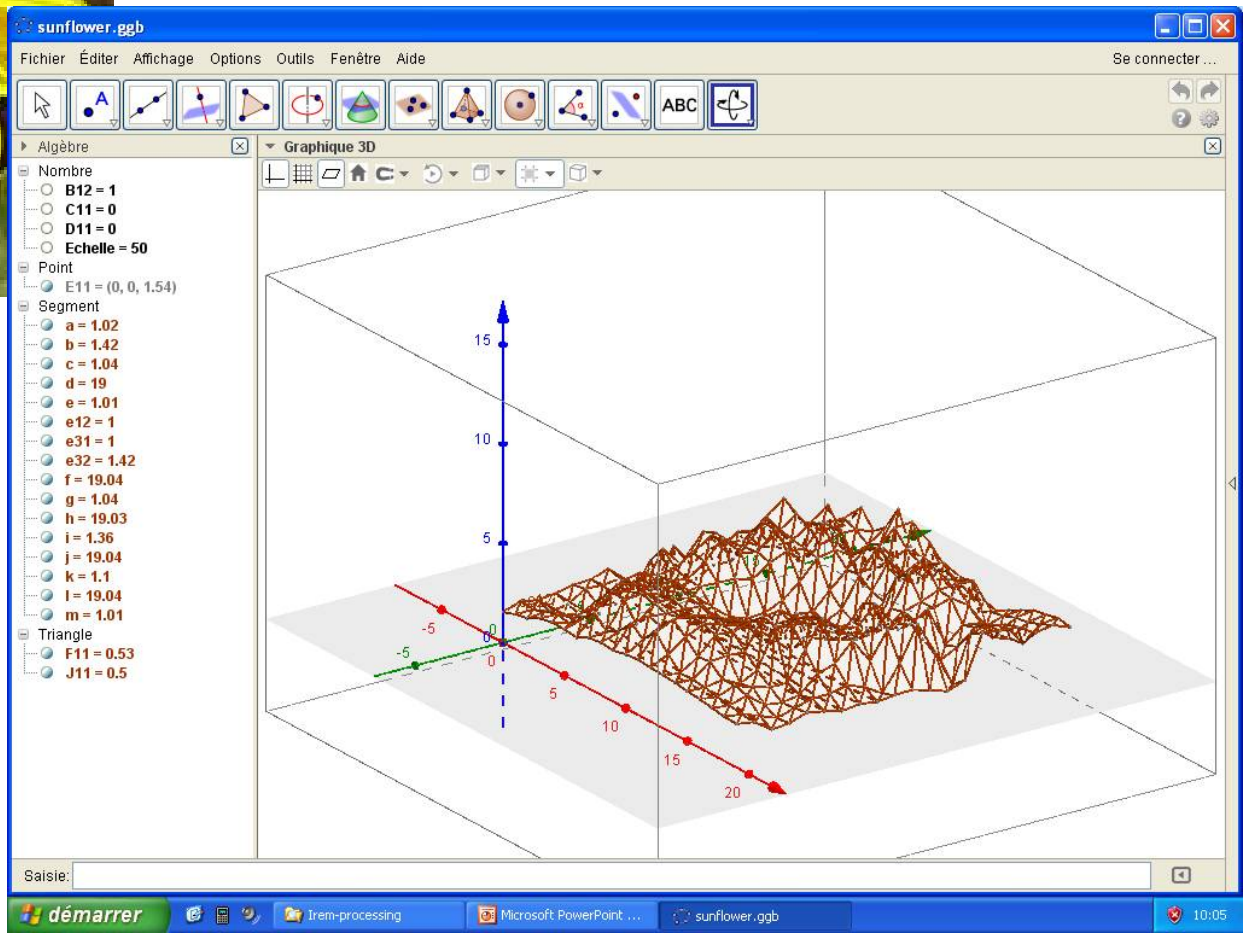
Lecture

numerisation_NB

taille texte = 9

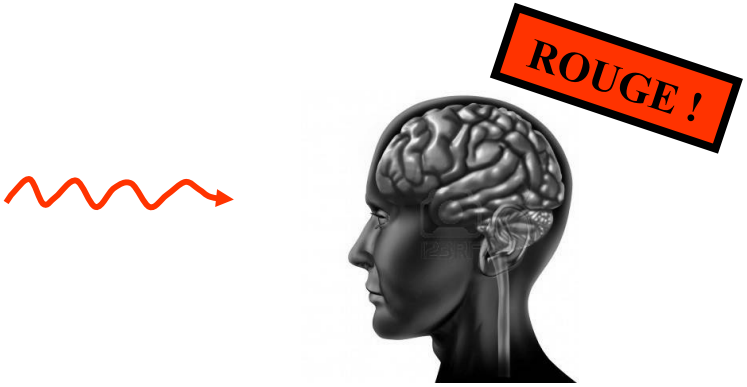
```
41  57  225 246 236 207 176 172 183 186 198 211 217 214 210 206 203 201 198 192 186 181 179 179 179
42  29  99  237 245 243 235 202 179 176 178 191 209 219 218 213 206 202 204 206 203 195 187 183 181
34  34  26  84  235 245 242 244 239 225 204 182 174 181 195 207 204 197 193 194 194 196 197 188 177
31  33  32  27  90  231 246 243 238 243 246 243 225 203 180 170 174 178 182 182 188 188 190 183 170
32  32  34  40  33  83  218 246 241 233 233 236 242 244 238 222 198 177 168 167 168 172 176 177 174
36  35  36  39  43  37  62  191 249 241 229 226 227 231 235 238 239 238 224 206 184 171 163 161 165
38  40  40  41  43  46  42  49  156 247 246 236 226 222 219 225 226 226 231 239 243 238 220 196 177
39  43  46  52  46  48  50  50  45  100 193 240 247 236 221 214 212 215 214 213 216 226 236 243 239
42  44  52  51  51  49  51  50  55  46  49  82  160 234 247 241 226 216 209 203 200 197 197 200 206
44  48  54  52  48  47  49  45  52  55  52  51  55  87  161 230 247 245 236 220 206 199 195 191 188
42  52  55  52  51  53  47  43  49  53  55  56  61  58  62  79  124 168 215 244 238 221 209 201 206
43  50  56  58  58  63  57  46  51  54  52  57  58  52  55  51  44  44  56  96  191 241 227 213 206
42  45  47  51  58  66  64  55  48  54  53  56  51  48  45  41  38  38  42  82  173 215 206 202 205
35  94  159 120 97  83  64  47  40  45  51  48  43  43  61  80  102 148 193 223 213 205 203 198 196
41  70  90  147 199 204 178 148 110 73  48  50  96  155 196 209 214 215 203 190 184 186 192 203 215
30  37  42  47  66  103 153 191 195 188 179 174 202 201 191 185 181 180 184 188 201 218 223 225 222
75  35  46  51  47  48  47  67  157 202 196 186 179 178 179 178 187 196 206 220 223 220 216 211 207
180 115 47  45  46  41  56  143 212 205 193 186 188 197 208 215 219 220 219 214 207 198 195 195 201
178 188 133 52  35  81  192 228 214 221 231 237 240 237 232 223 217 209 203 199 205 213 226 238 242
166 172 169 139 130 220 231 229 241 244 243 239 234 220 211 209 211 216 225 235 243 244 244 244 243
145 150 173 227 237 236 241 244 237 232 226 220 219 224 234 241 243 245 245 244 244 243 243 243 242
139 164 229 238 242 243 238 231 223 221 229 235 242 245 245 244 244 244 244 243 244 242 241 241 241
161 225 238 239 233 228 228 231 238 243 243 244 244 243 242 242 244 244 243 242 242 240 238 238 236
215 228 227 228 234 240 244 244 245 242 242 242 243 244 243 240 238 237 235 235 236 235 235 236 235
217 223 237 243 244 243 242 240 240 241 240 239 236 234 233 234 233 233 232 234 237 238 240 241 241
```

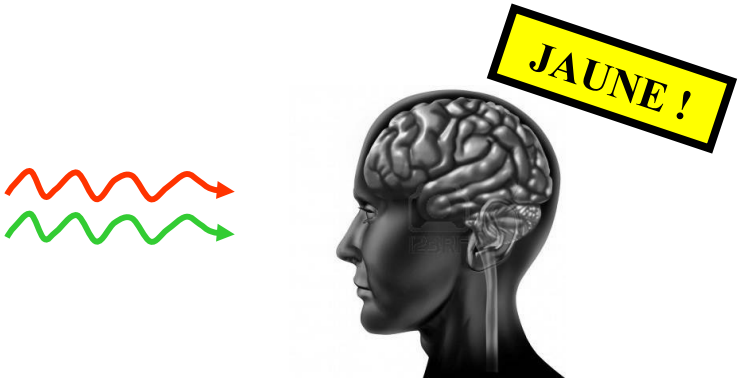


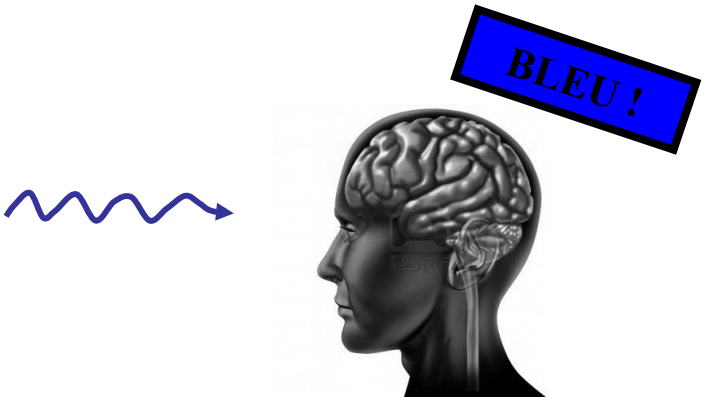

sunflower.ggb
Fichier Éditer Affichage Options Outils Fenêtre Aide
Se connecter ...
ABC
Algèbre
Nombre
B12 = 1
C11 = 0
D11 = 0
Echelle = 50
Point
E11 = (0, 0, 1.54)
Segment
a = 1.02
b = 1.42
c = 1.04
d = 19
e = 1.01
e12 = 1
e31 = 1
e32 = 1.42
f = 19.04
g = 1.04
h = 19.03
i = 1.36
j = 19.04
k = 1.1
l = 19.04
m = 1.01
Triangle
F11 = 0.53
J11 = 0.5
Graphique 3D
15
10
5
-5
0
-5
0
5
10
15
20
Saisie:
démarrer
Irem-processing
Microsoft PowerPoint ...
sunflower.ggb
10:05

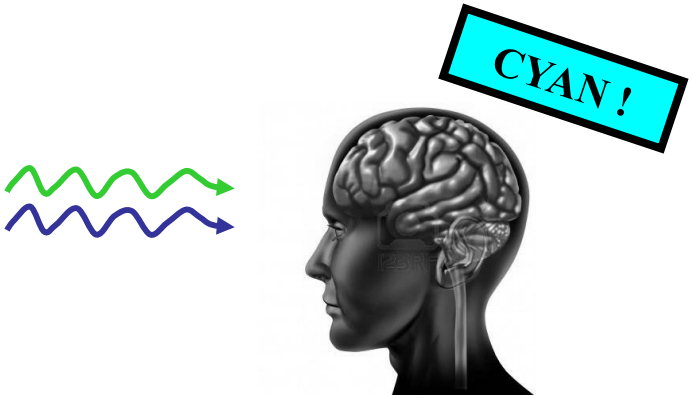
Lecture

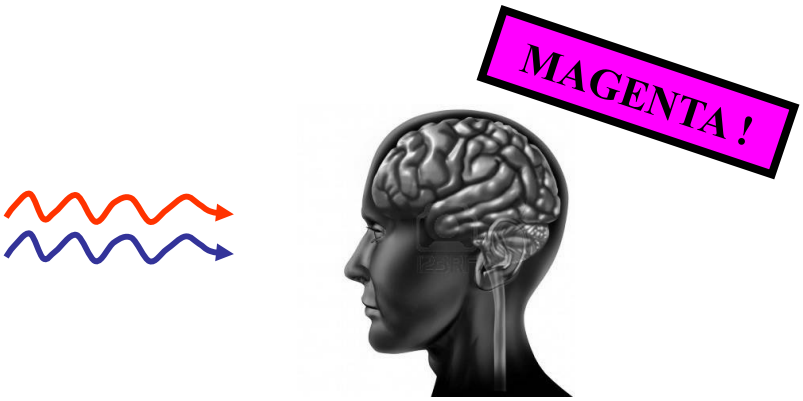
RESTITUTION DES COULEURS

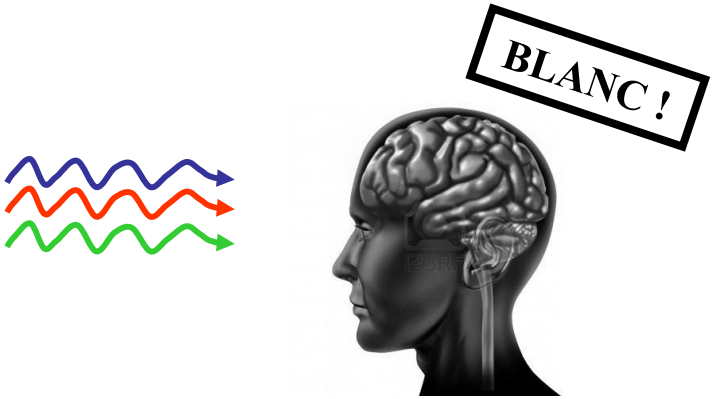

ROUGE !


JAUNE !


BLEU !


CYAN !


MAGENTA !


BLANC !

SYNTHESE ADDITIVE

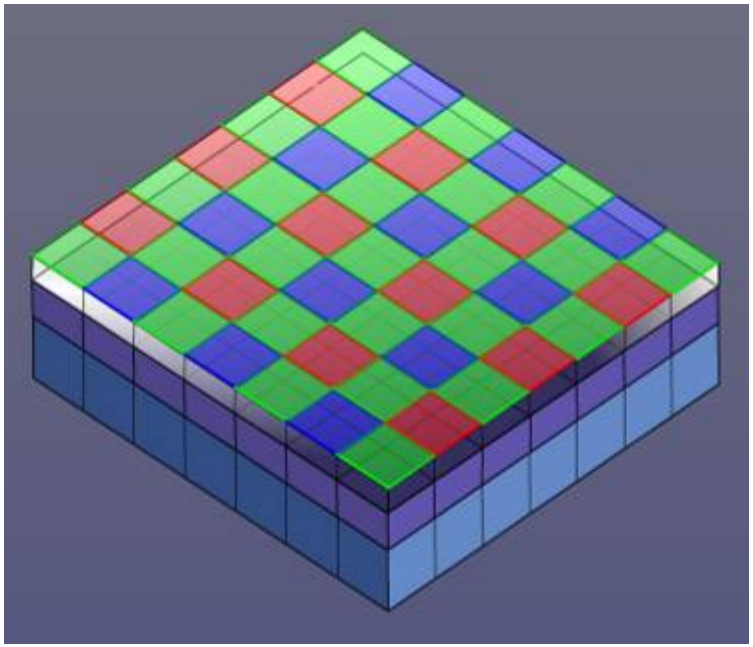
R + V = JAUNE

R + B = MAGENTA

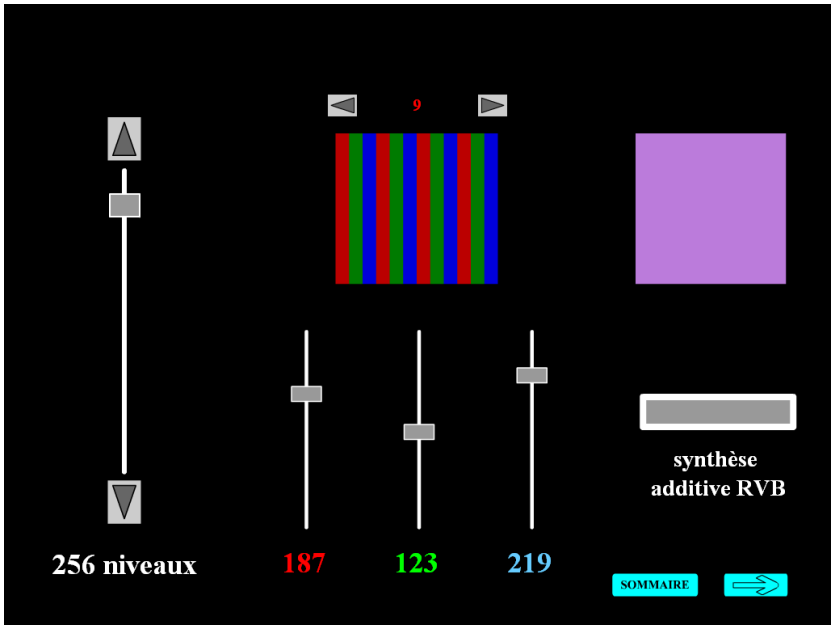
V + B = CYAN

R + V +B = BLANC

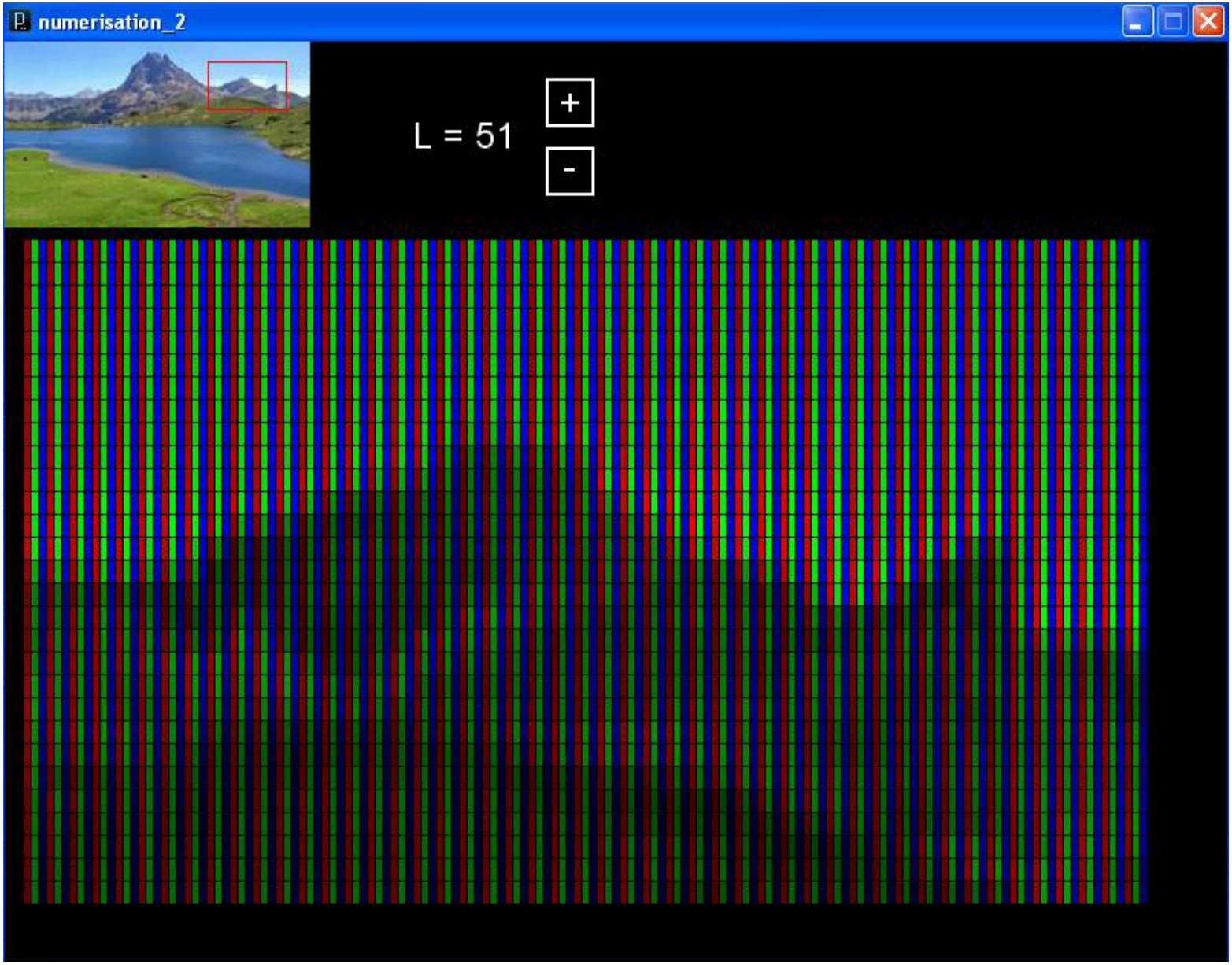
Ce principe est utilisé pour la restitution des couleurs sur tous les écrans plats (ordinateur, téléphone , TV…)



Visualisation des pixels sur écran

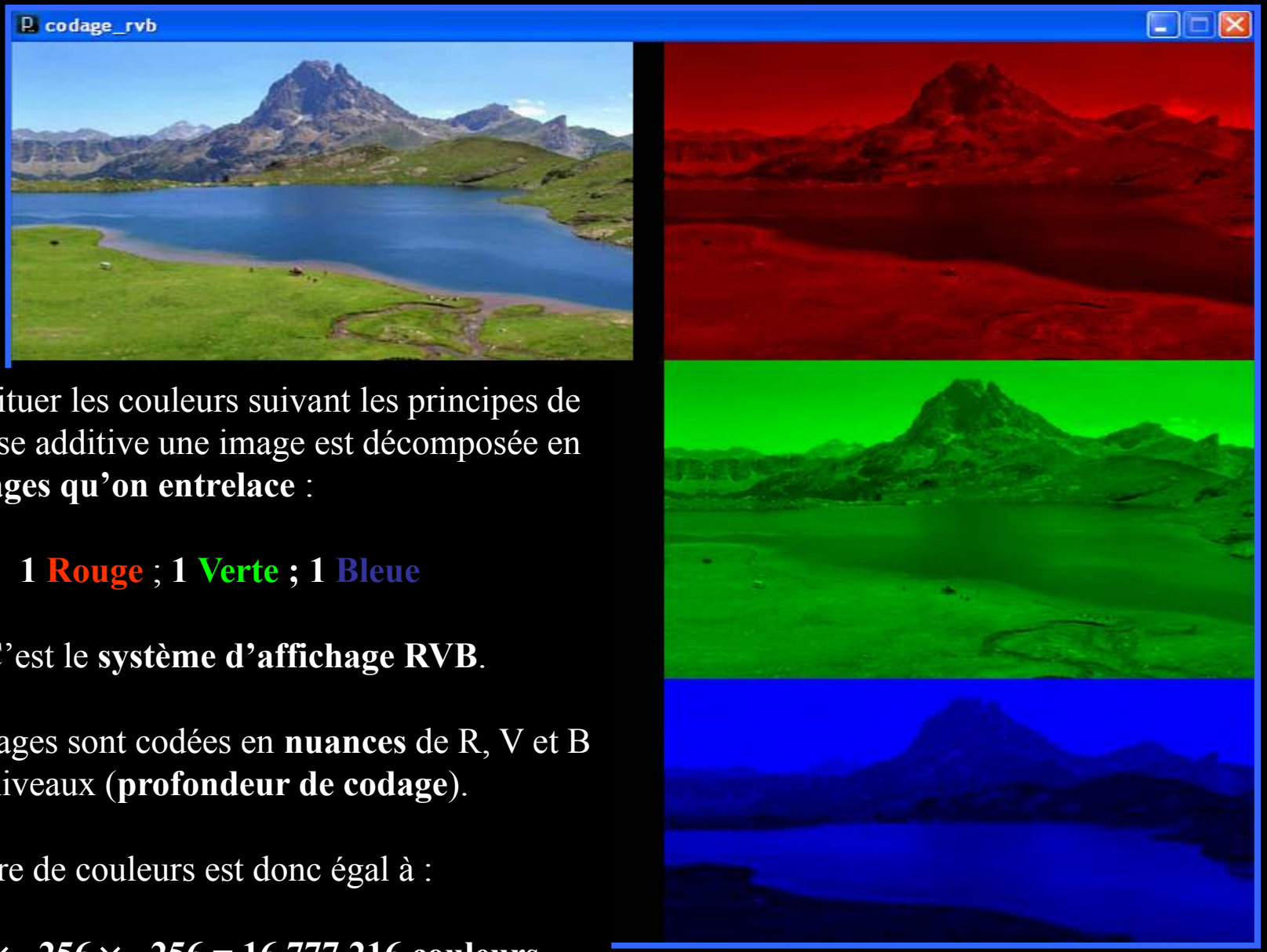


Lecture



Lecture

Lecture



Pour restituer les couleurs suivant les principes de la synthèse additive une image est décomposée en **trois images qu'on entrelace** :

1 Rouge ; 1 Verte ; 1 Bleue

C'est le **système d'affichage RVB**.

Ces 3 images sont codées en **nuances** de R, V et B sur 256 niveaux (**profondeur de codage**).

Le nombre de couleurs est donc égal à :

256 × 256 × 256 = 16 777 216 couleurs

Environ 16,8 millions de couleurs !

Lecture

Traitements numériques des images

Par définition, un signal est le **support de l'information** émise par une source et destinée à un récepteur. Il transporte sur les réseaux l'information, la **parole** ou **l'image**… Le traitement qu'il subit a pour but **d'extraire des informations**, de modifier le message qu'il transporte ou de **l'adapter aux moyens de transmission**.

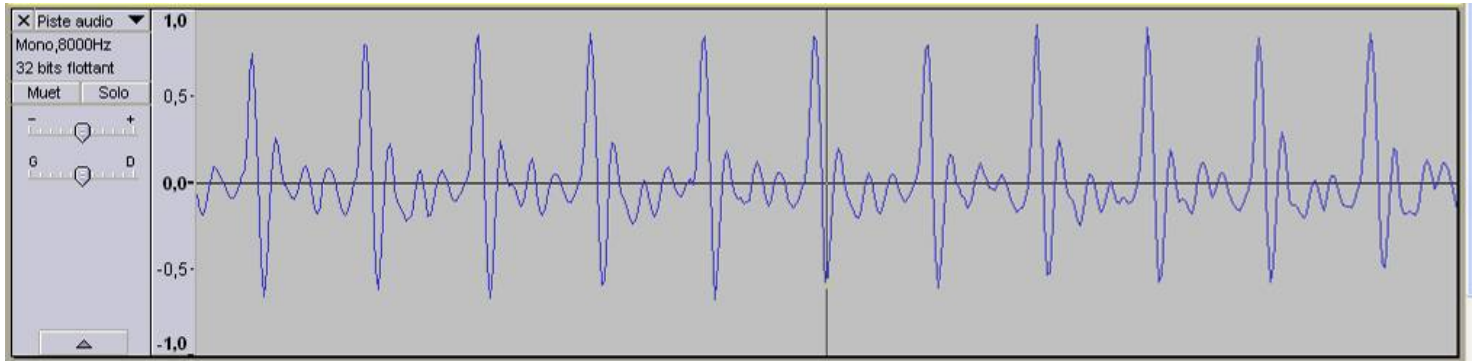
C'est là qu'interviennent les techniques numériques. En effet, si l'on imagine de **substituer au signal un ensemble de nombres** qui représentent sa grandeur ou amplitude à des instants convenablement choisis, le traitement, même dans sa forme la plus élaborée, se ramène à une **séquence d'opérations logiques et arithmétiques** sur cet ensemble de nombres, associées à des mises en mémoire.

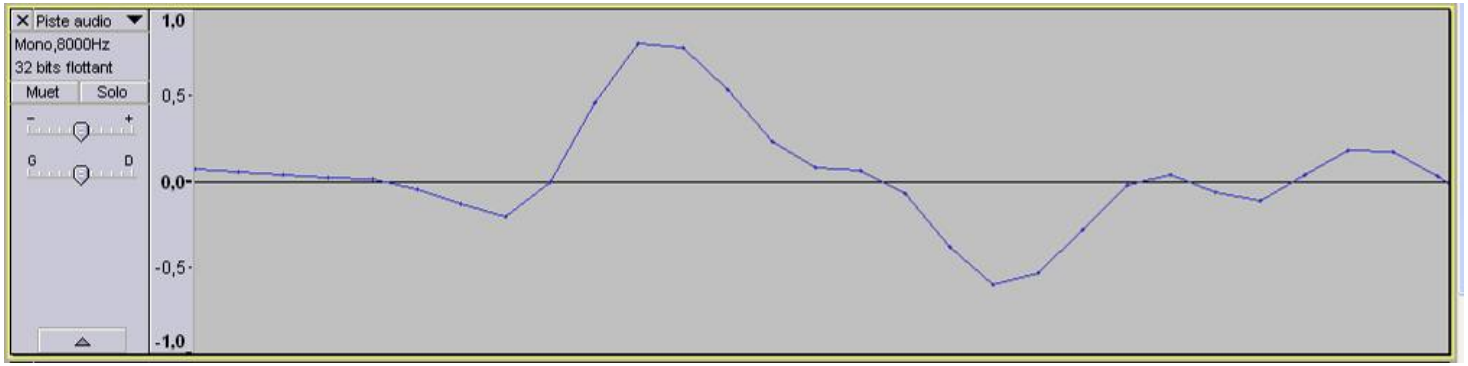
Le **traitement numérique** du signal désigne **l'ensemble des opérations**, calculs arithmétiques et manipulations de nombres, qui sont effectués sur un signal à traiter, représenté par une suite ou un ensemble de nombres, **en vue de fournir une autre suite ou un autre ensemble de nombres**, qui représentent le signal traité. Les fonctions les plus variées sont réalisables de cette manière, comme l'analyse spectrale, le filtrage, le transcodage, la modulation, la détection, l'estimation et l'extraction de paramètres.

Maurice Bellanger – traitement numérique du signal -Dunod

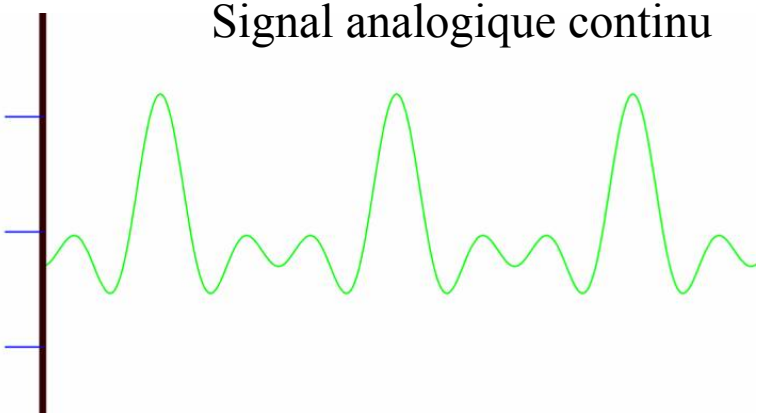
La numérisation du signal - Échantillonnage et codage

La conversion d'un signal **analogique** sous forme **numérique** implique une double approximation. D'une part, dans l'espace des temps, le signal fonction du temps $s(t)$ est remplacé par ses valeurs $s(n\mathrm{T})$ à des instants multiples entiers d'une durée T : c'est l'opération **d'échantillonnage**. D'autre part, dans l'espace des amplitudes, chaque valeur $s(n\mathrm{T})$ est approchée par un multiple entier d'une quantité élémentaire q : c'est l'opération de **quantification**. La valeur approchée ainsi obtenue est ensuite **associée à un nombre : c'est le codage**.

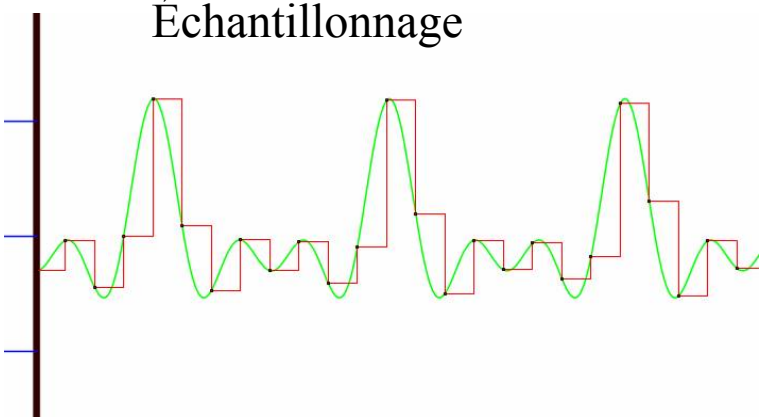




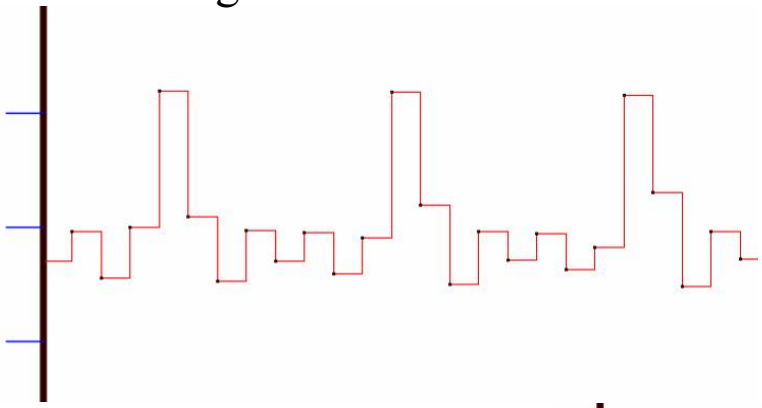
Signal analogique continu



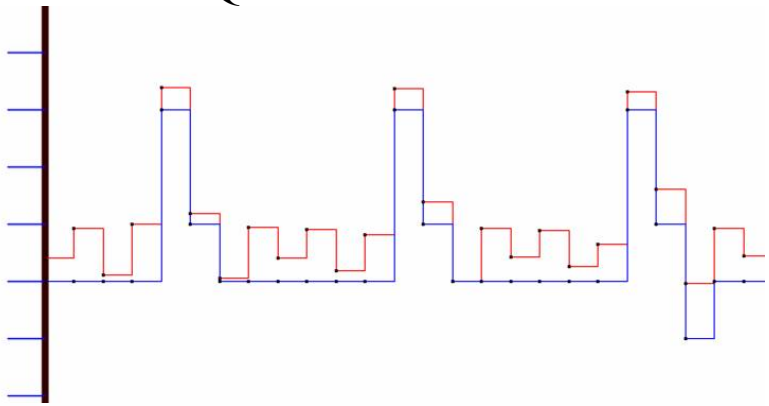
Échantillonnage



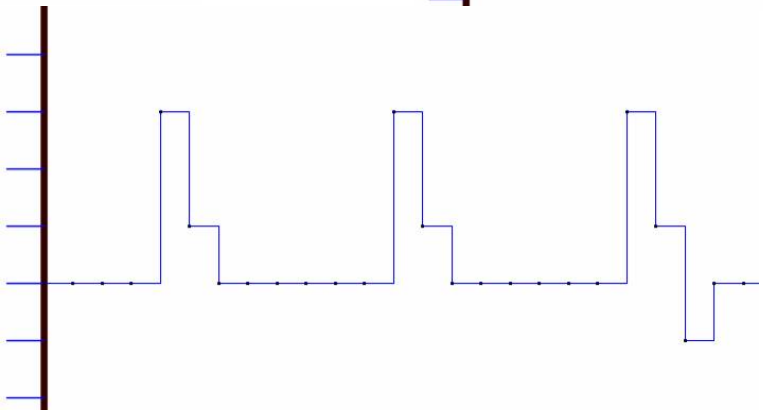
Signal échantillonné discret



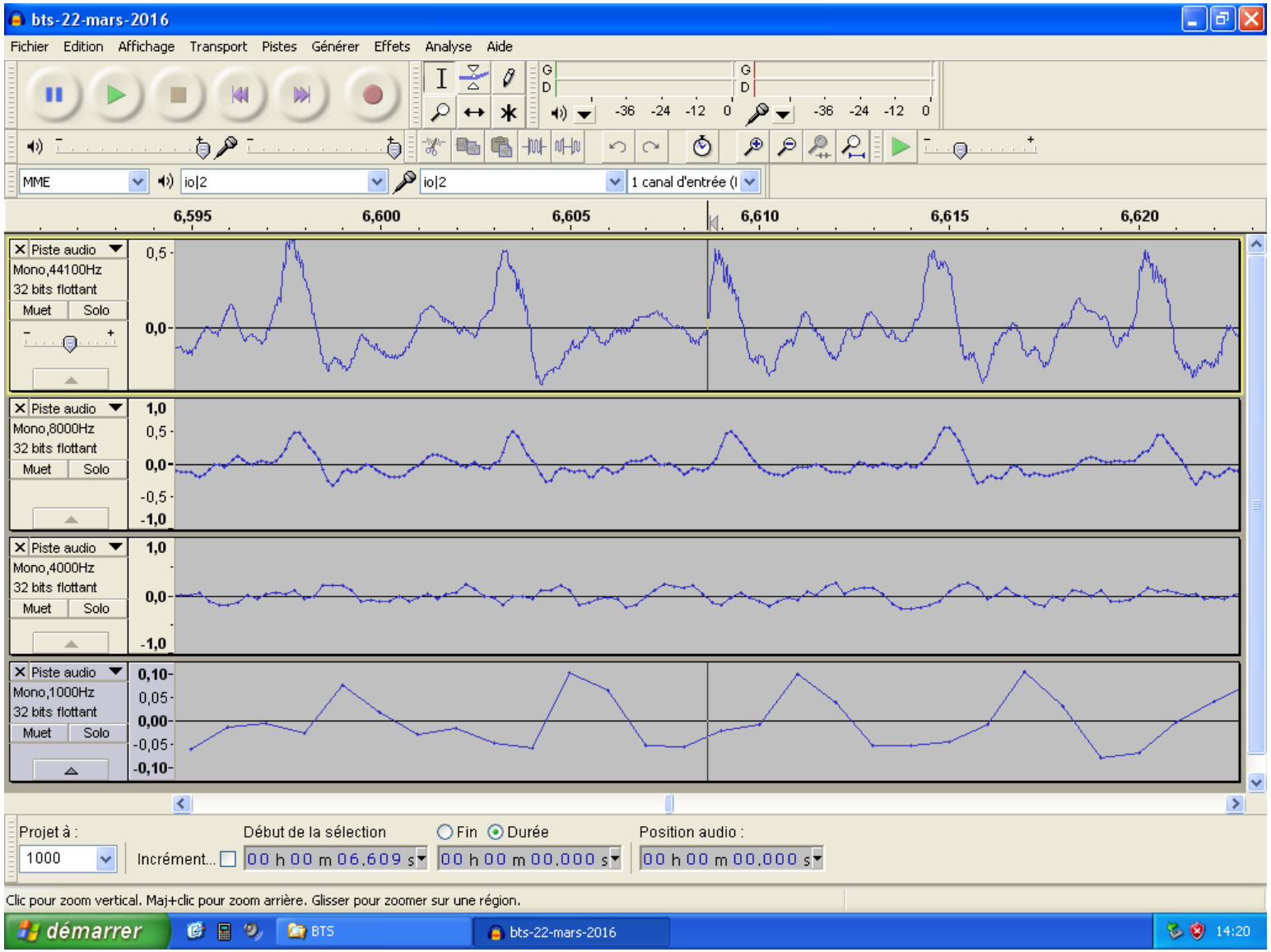
Quantification

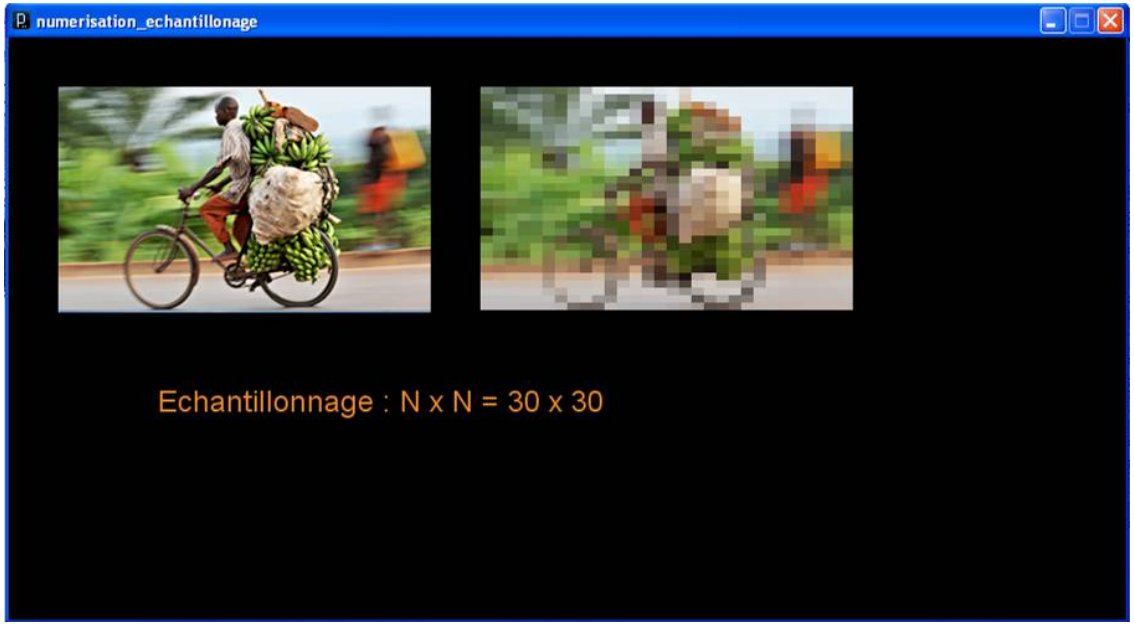


Signal codé discret



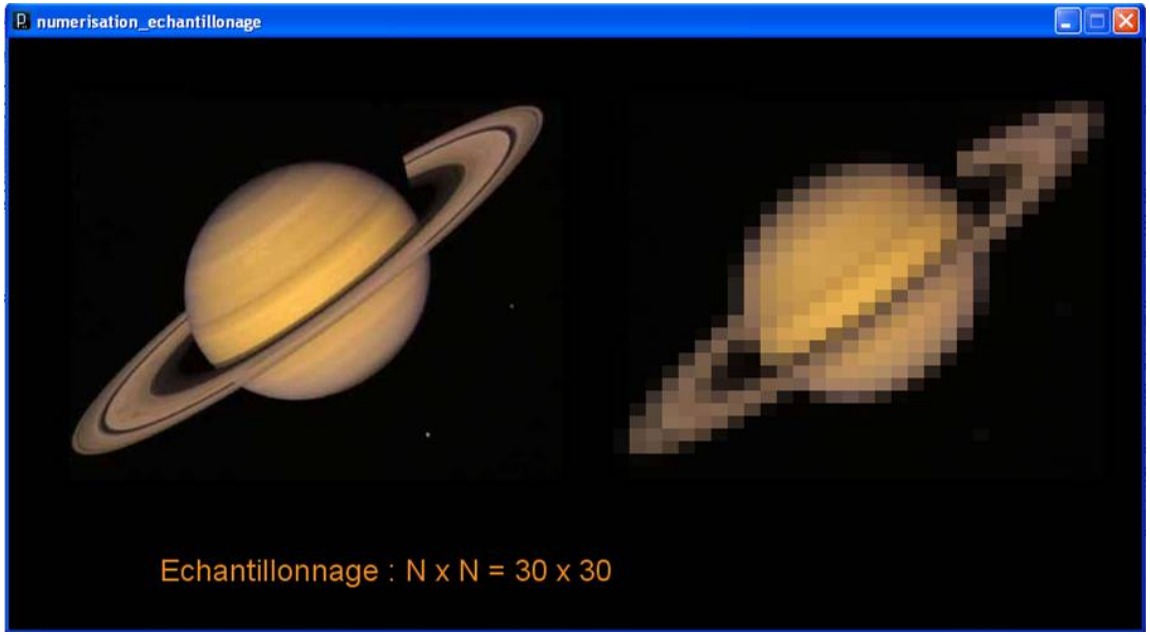
Influence de la fréquence d'échantillonnage



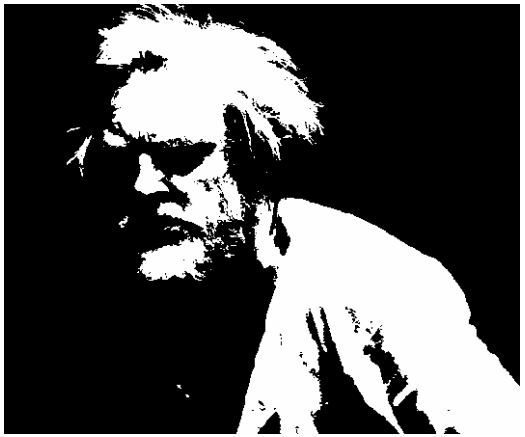


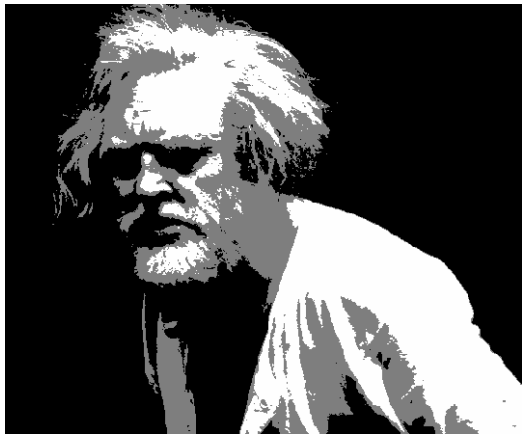
Lecture

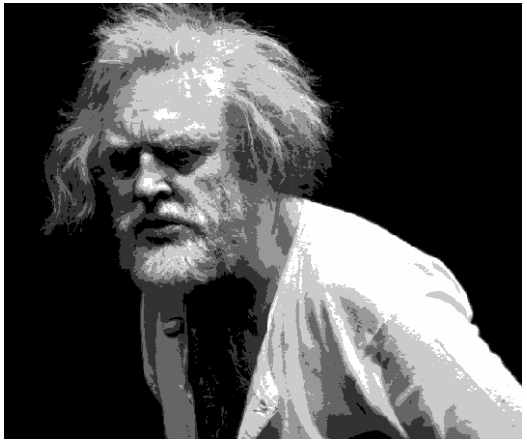
Influence de la fréquence d'échantillonnage

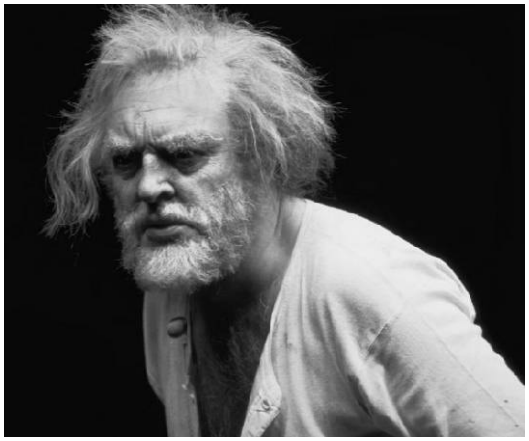


Influence de la quantification







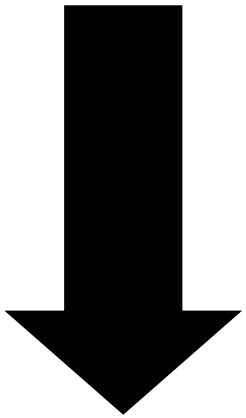


Influence de la quantification

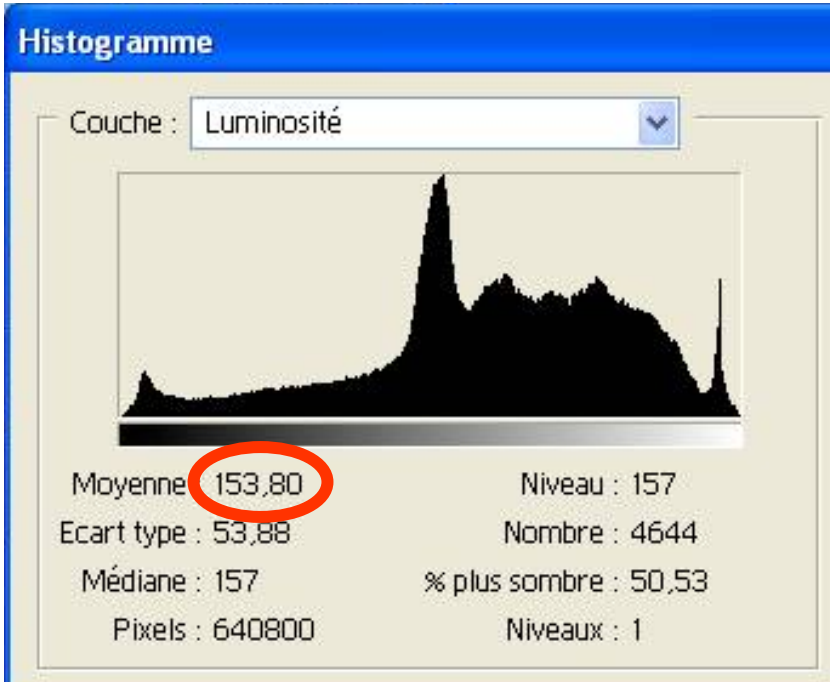
Quelques exemples

de traitements

des images numériques



1. Luminosité d'une image



On peut définir la luminosité d'une image à partir de la **valeur moyenne des luminosités** (ou niveaux de gris) de l'ensemble des pixels de l'image.

Plus cette valeur moyenne est élevée et plus l'image est lumineuse !

Histogramme

Couche : Luminosité

Moyenne : 197,71
Ecart type : 49,85
Médiane : 204
Pixels : 640800
Niveau : 204
Nombre : 4644
% plus sombre : 50,53
Niveaux : 1

Histogramme

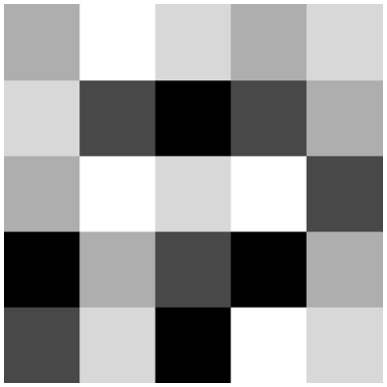
Couche : Luminosité

Moyenne : 103,43	Niveau : 105
Ecart type : 50,17	Nombre : 4644
Médiane : 105	% plus sombre : 50,53
Pixels : 640800	Niveaux : 1

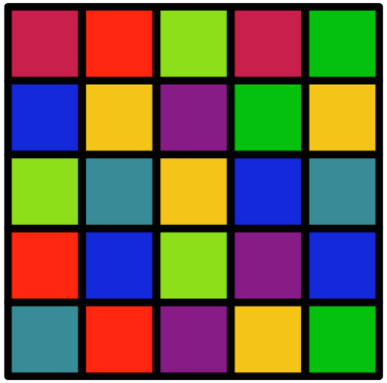
2. Contraste d’une image

La notion de contraste est intimement liée à la **perception d'une différence**, d'un écart, d'une disparité, voire d'une opposition entre deux objets ou deux concepts. Il y a d'un côté le blanc et le noir, le chaud et le froid, le grand et le petit, le lourd et le léger, le fort et le faible, etc.

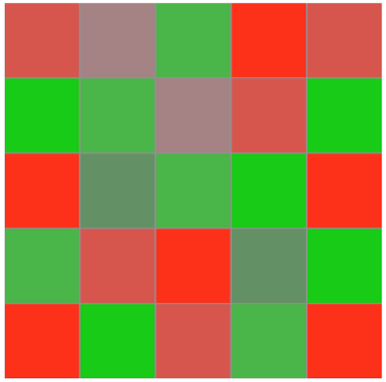
En photographie, on rencontre couramment divers types de contrastes, dont quelques uns peuvent être représentés par les petits schémas ci-dessous :



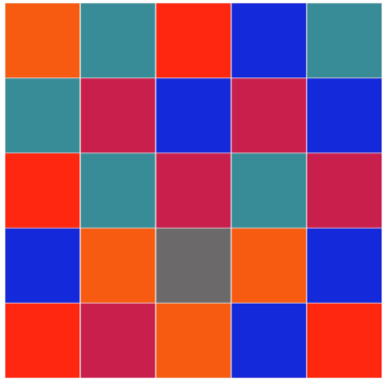
Luminosité



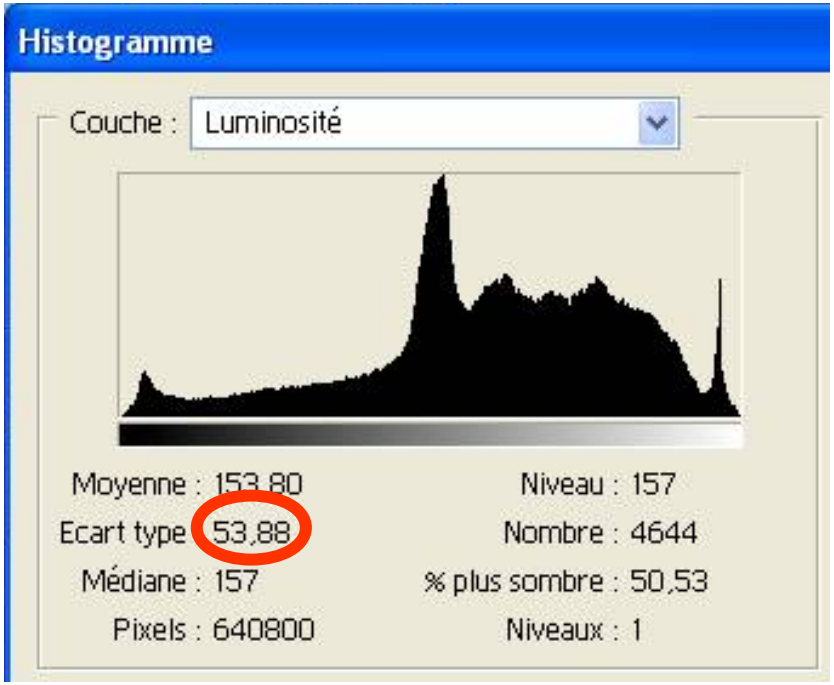
Couleurs



Couleurs complémentaires



tons chauds/froids

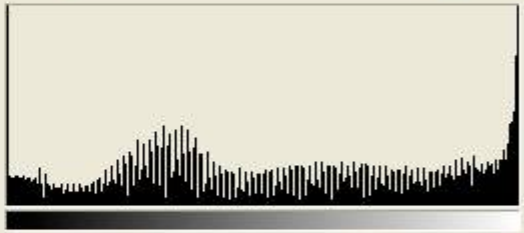


On peut définir le contraste à partir de la **valeur de l'écart type** des niveaux de gris de l'ensemble des pixels de l'image. Il s'agit de voir si les niveaux de luminosité sont répartis de façon homogène ou pas..

Plus l'écart type est grand et plus l'image est contrastée !

Histogramme

Couche : Luminosité



Moyenne : 149,15
Ecart type : 95,08
Médiane : 163
Pixels : 640800

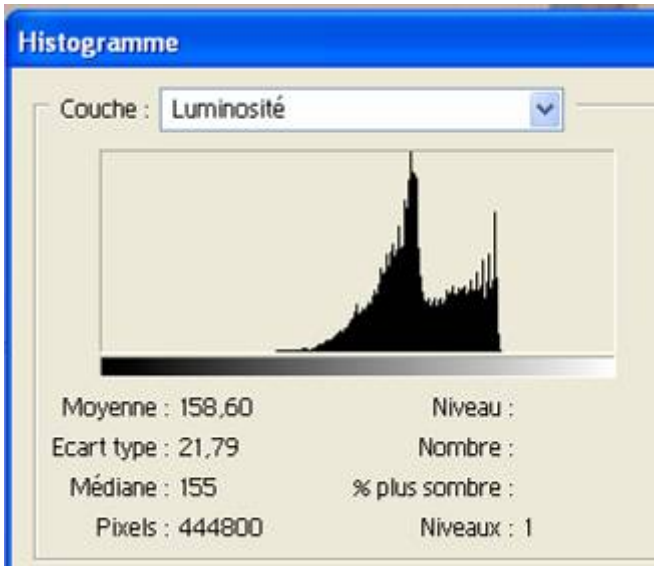
Niveau :
Nombre :
% plus sombre :
Niveaux : 1

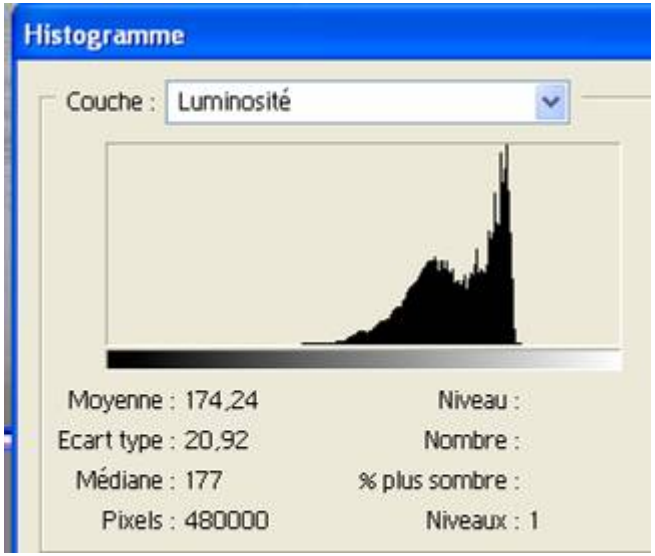
Histogramme

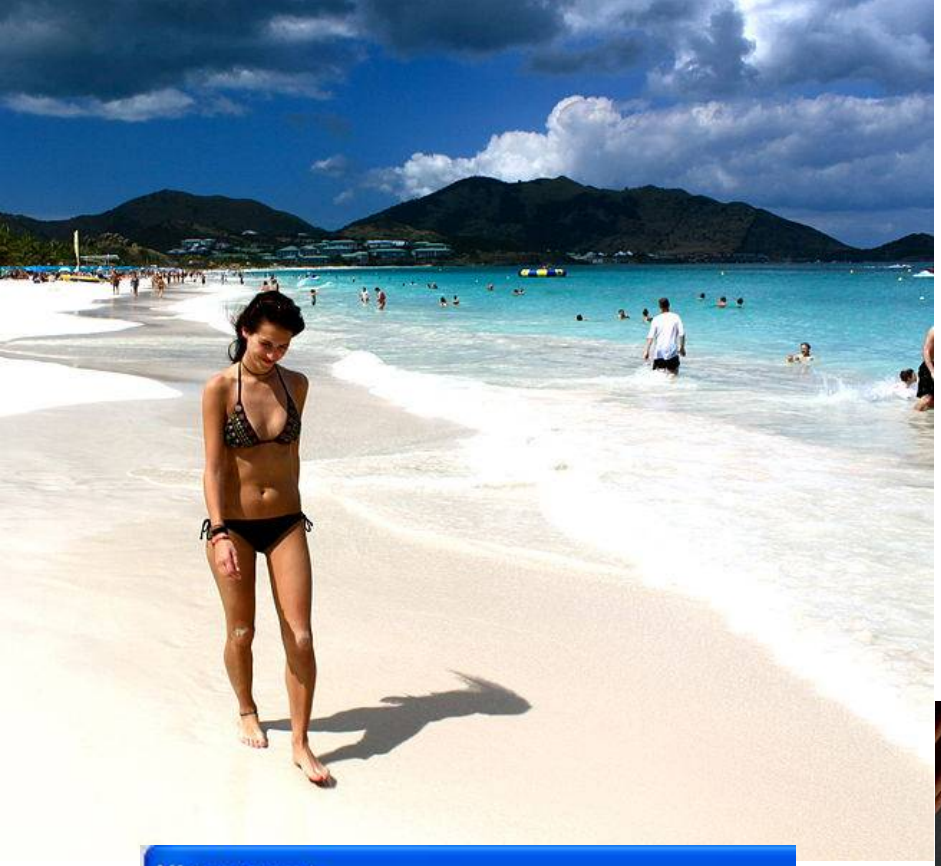
Couche : Luminosité

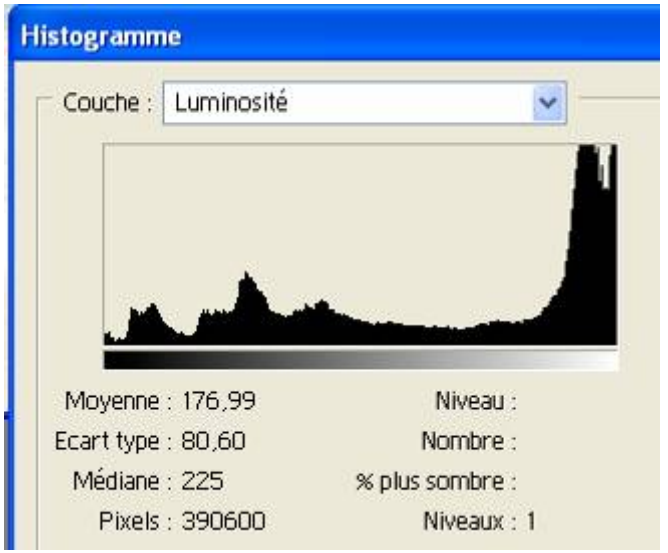
Moyenne : 153,93
Ecart type : 18,37
Médiane : 155
Pixels : 640800

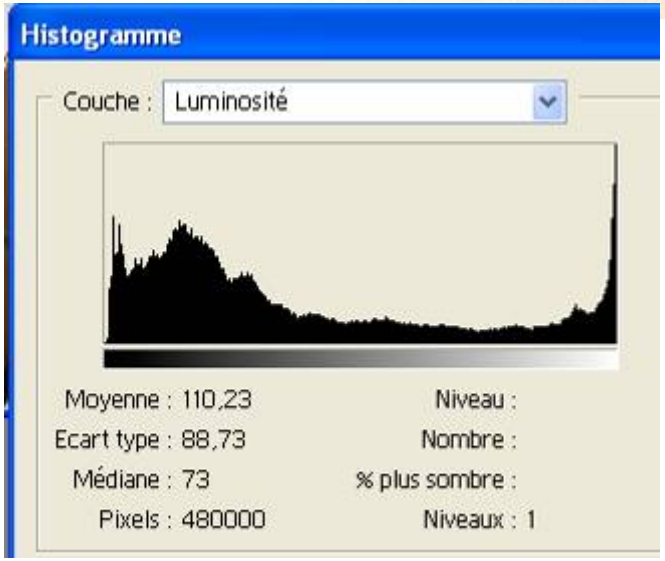
Niveau :
Nombre :
% plus sombre :
Niveaux : 1


Histogramme
Couche : Luminosité
Moyenne : 158,60
Ecart type : 21,79
Médiane : 155
Pixels : 444800
Niveau :
Nombre :
% plus sombre :
Niveaux : 1

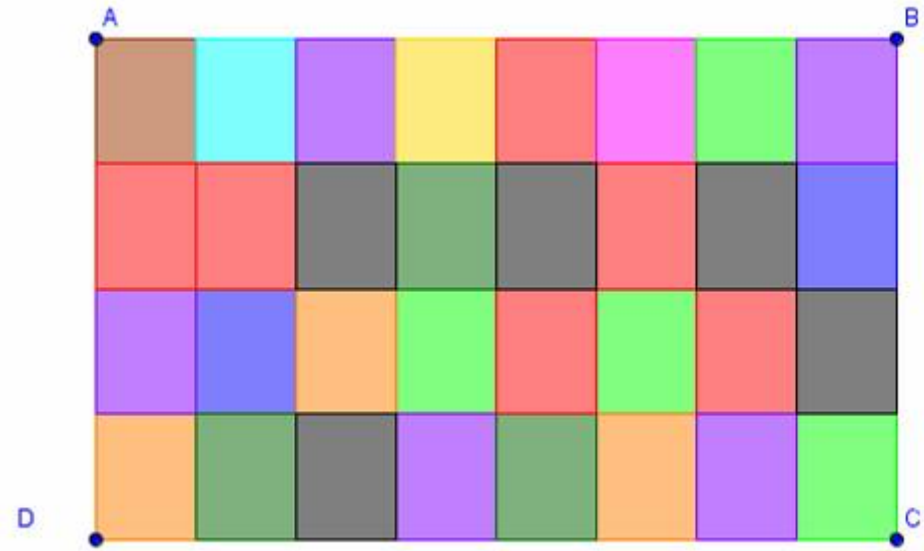

Histogramme
Couche : Luminosité
Moyenne : 174,24
Ecart type : 20,92
Médiane : 177
Pixels : 480000
Niveau :
Nombre :
% plus sombre :
Niveaux : 1

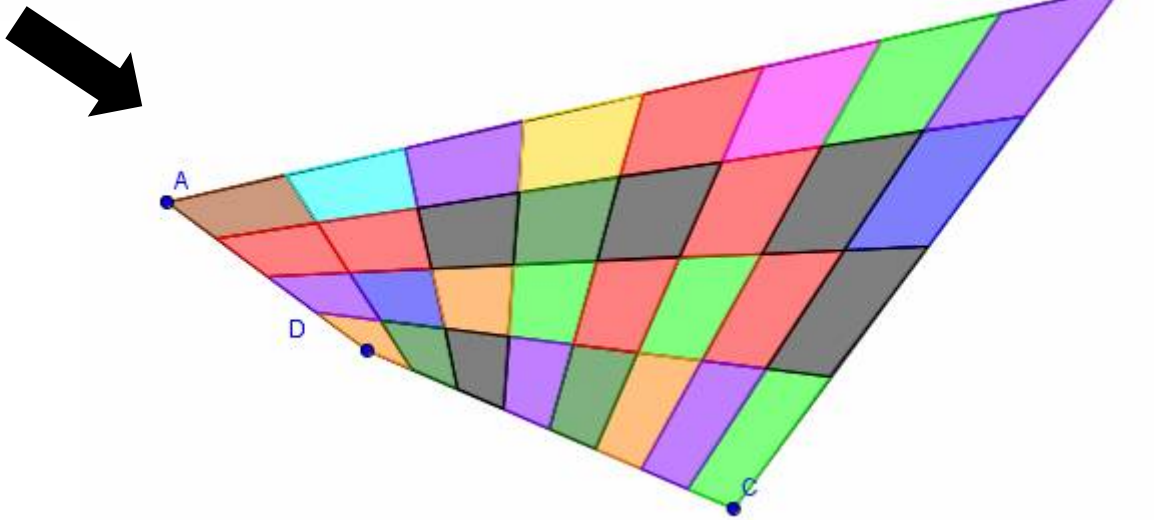



Histogramme
Couche : Luminosité
Moyenne : 176,99
Ecart type : 80,60
Médiane : 225
Pixels : 390600
Niveau :
Nombre :
% plus sombre :
Niveaux : 1


Histogramme
Couche : Luminosité
Moyenne : 110,23
Ecart type : 88,73
Médiane : 73
Pixels : 480000
Niveau :
Nombre :
% plus sombre :
Niveaux : 1

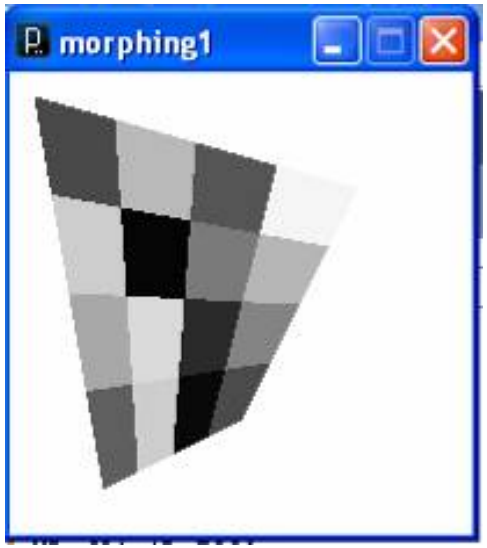
3. le morphing

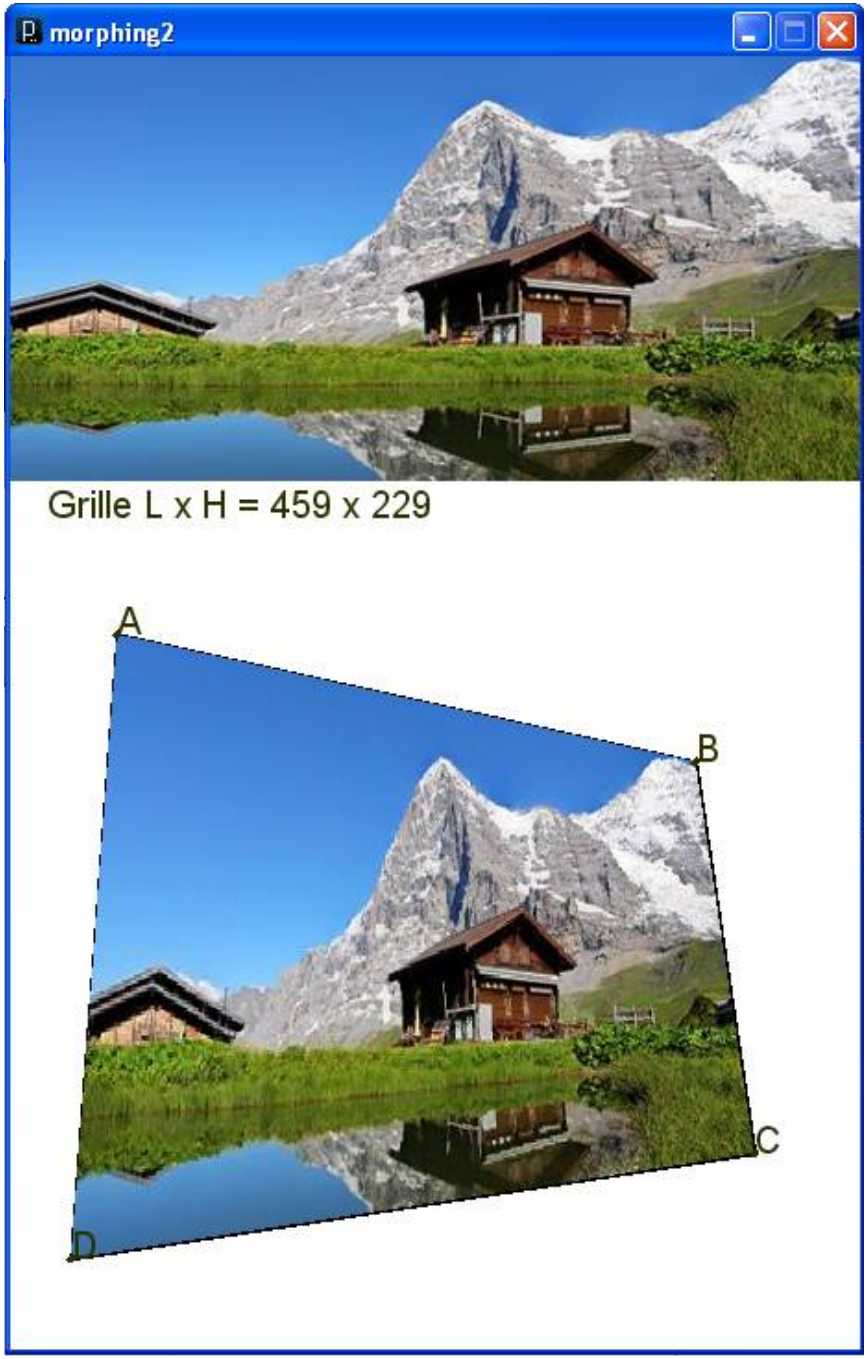




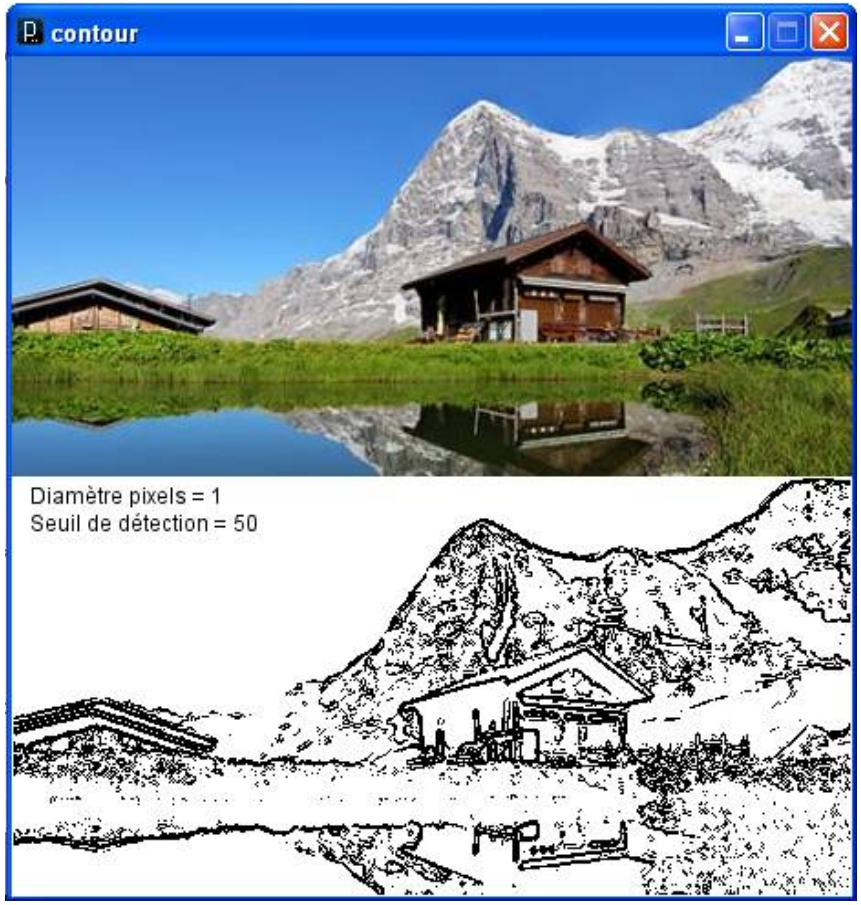
Lecture

Lecture



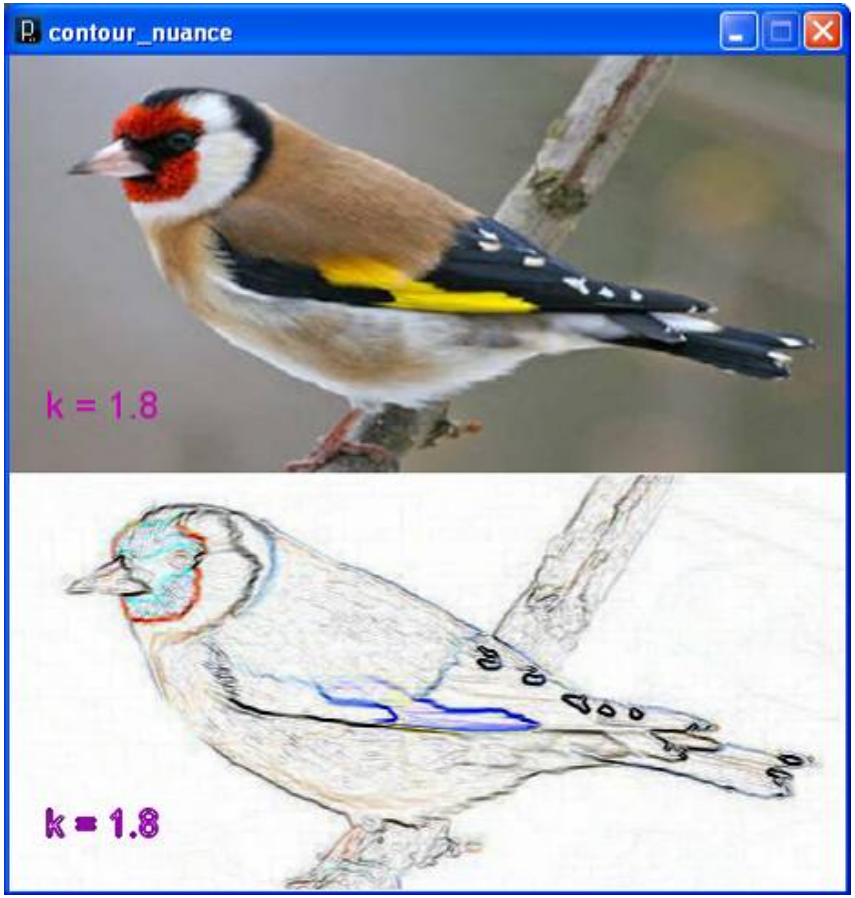


4. Détection de contours



Lecture

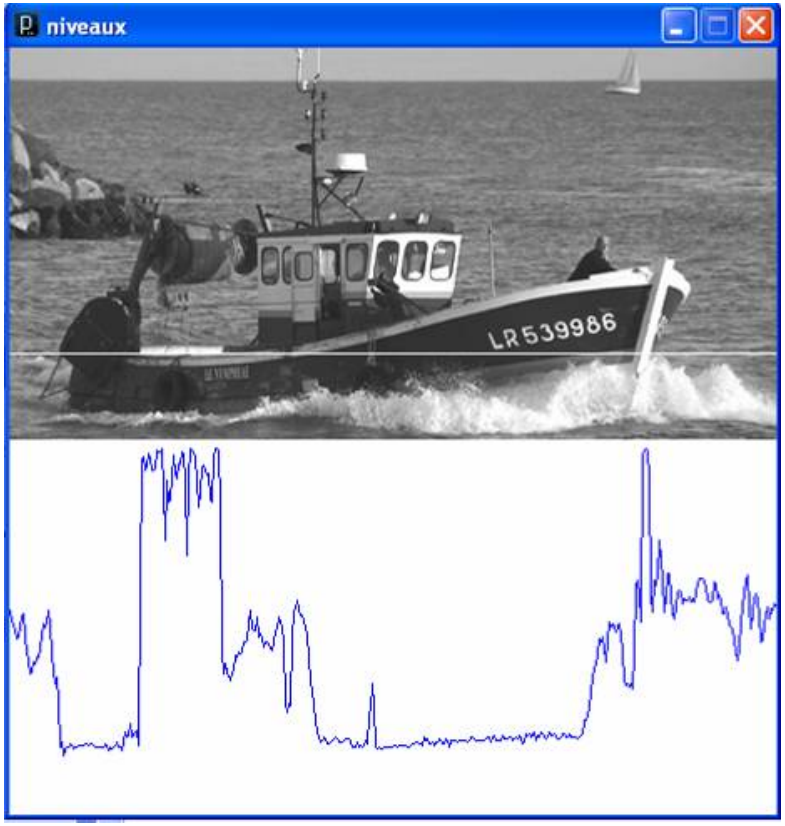
Lecture

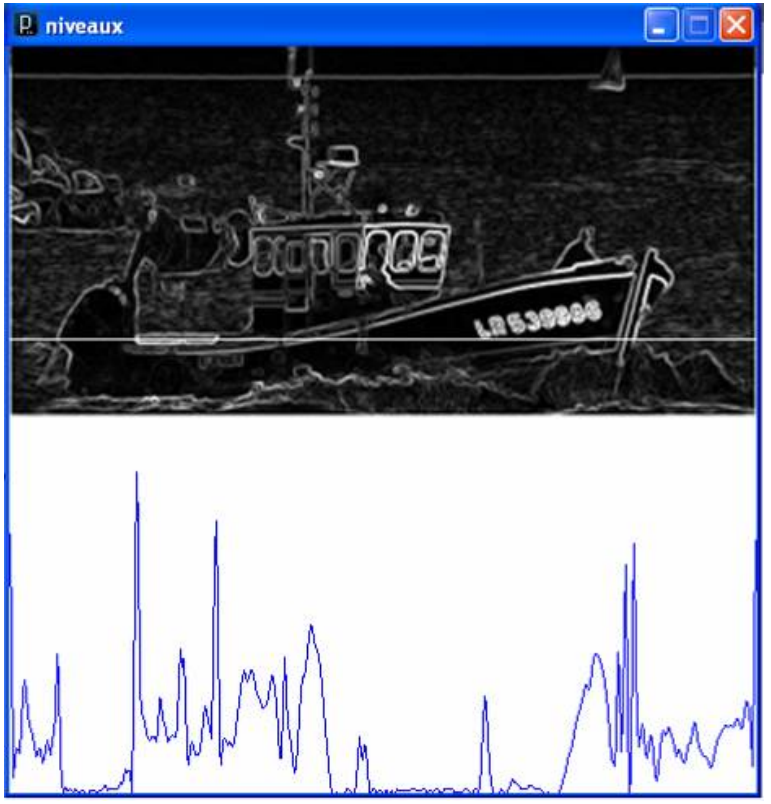


Principe…

Lecture

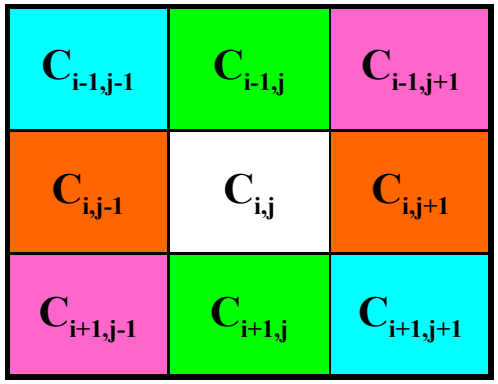
Principe avec une image N&B…





Analyse du code

On calcule les **taux de variations** verticaux, horizontaux et en diagonale (d1, d2, d3 ,d4), dont on prend ensuite la valeur absolue.



deriv | Processing 1.5.1

File Edit Sketch Tools Help

deriv

```
for (int i=1;i<L-1;i=i+1) {
  for (int j=1 ; j<H-1 ; j=j+1) {
    noStroke();
    float d1 = red( c[i-1][j])- red(c[i+1][j]);
    float d2 = red( c[i][j-1])- red(c[i][j+1]);
    float d3 = red( c[i-1][j-1])- red(c[i+1][j+1]);
    float d4 = red( c[i-1][j+1])- red(c[i+1][j-1]);

    if (d1<0) {
      d1 = -d1;
    }
    if (d2<0) {
      d2 = -d2;
    }
    if (d3<0) {
      d3 = -d3;
    }
    if (d4<0) {
      d4 = -d4;
    }

    //float niveauGris=255-(d1+d2+d3+d4)/3;
    float niveauGris=(d1+d2+d3+d4)/2;
    stroke(niveauGris, niveauGris, niveauGris);
    point(i, j+H);
  }
```

La détection de contour correspond à une « **dérivation** » de l'image, en 2D.

Lecture

5. Filtrage d'une image

Rappel :

Le développement en série de Fourier d'un signal périodique (période T) s'écrit en notation complexe :

$$s(t) = \underline{C}_0 + \sum_{n=1}^{\infty} \underline{C}_n e^{-j2\pi \times nf \times t}$$

$$n > 0 \Rightarrow \underline{C}_n(nf) = \frac{2}{T}\int_0^T s(t) e^{-j2\pi \times nf \times t} dt$$

$$n = 0 \Rightarrow \underline{C}_0 = \frac{1}{T}\int_0^T s(t) dt$$

Ou plus simplement, sous forme réelle :

$$s(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(2\pi \times nf \times t + \phi_n)$$

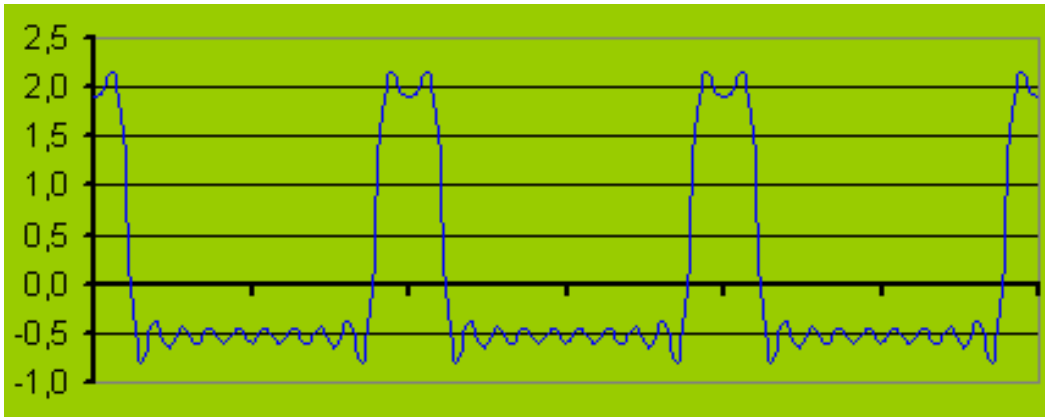
$$C_n = \left|\underline{C}_n\right|$$

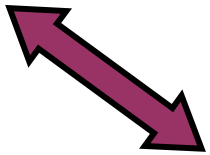
$$\phi_n = \arg(\underline{C}_n)$$

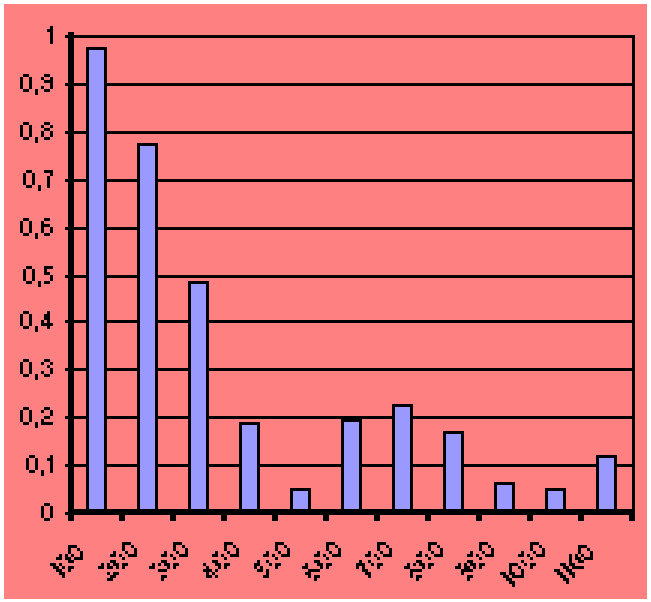
$$s(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(2\pi \times nf \times t + \phi_n)$$

Spectre de fréquence :

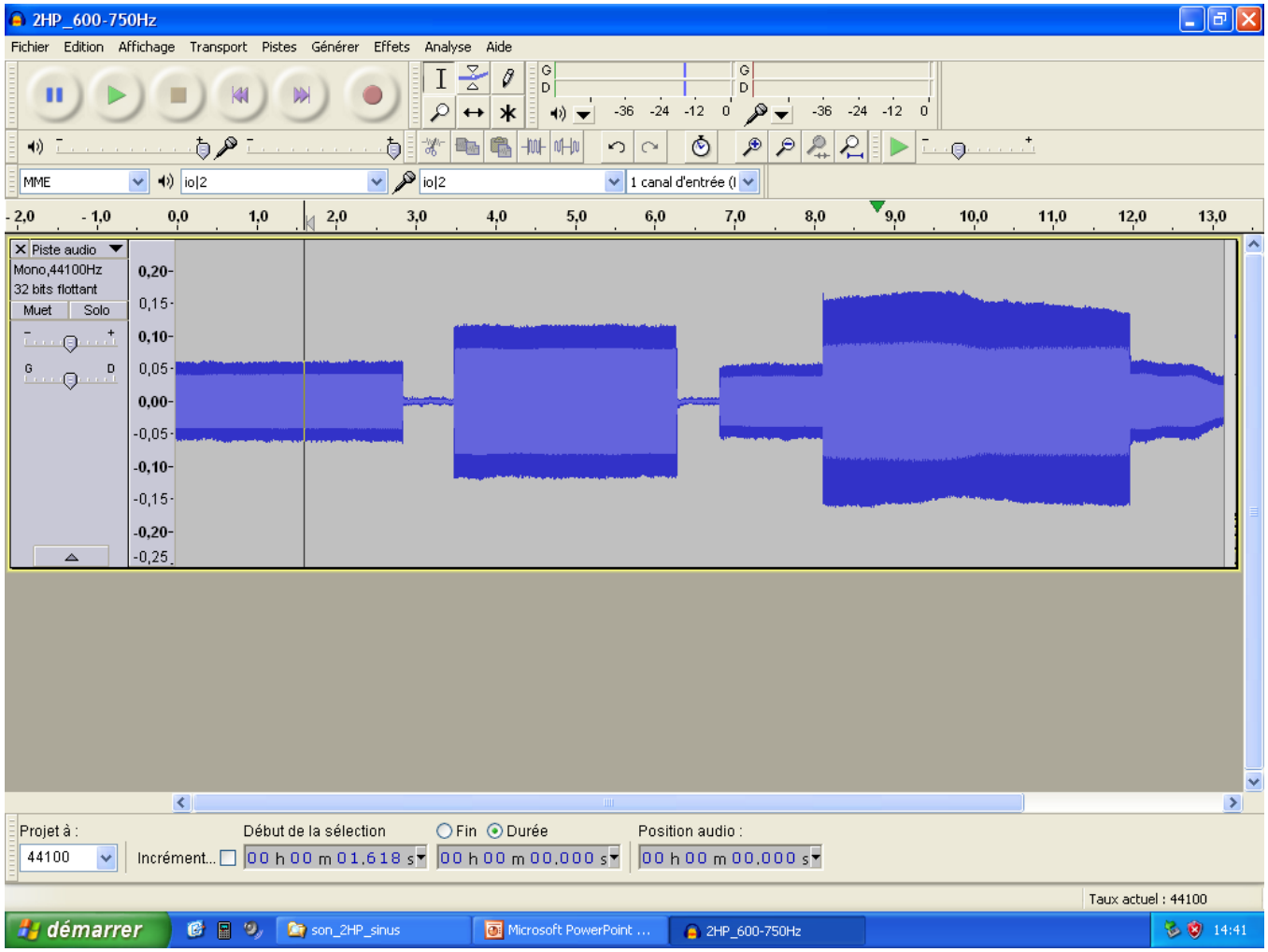
L'ensemble des amplitudes C_n forme le spectre de fréquences du signal s(t)



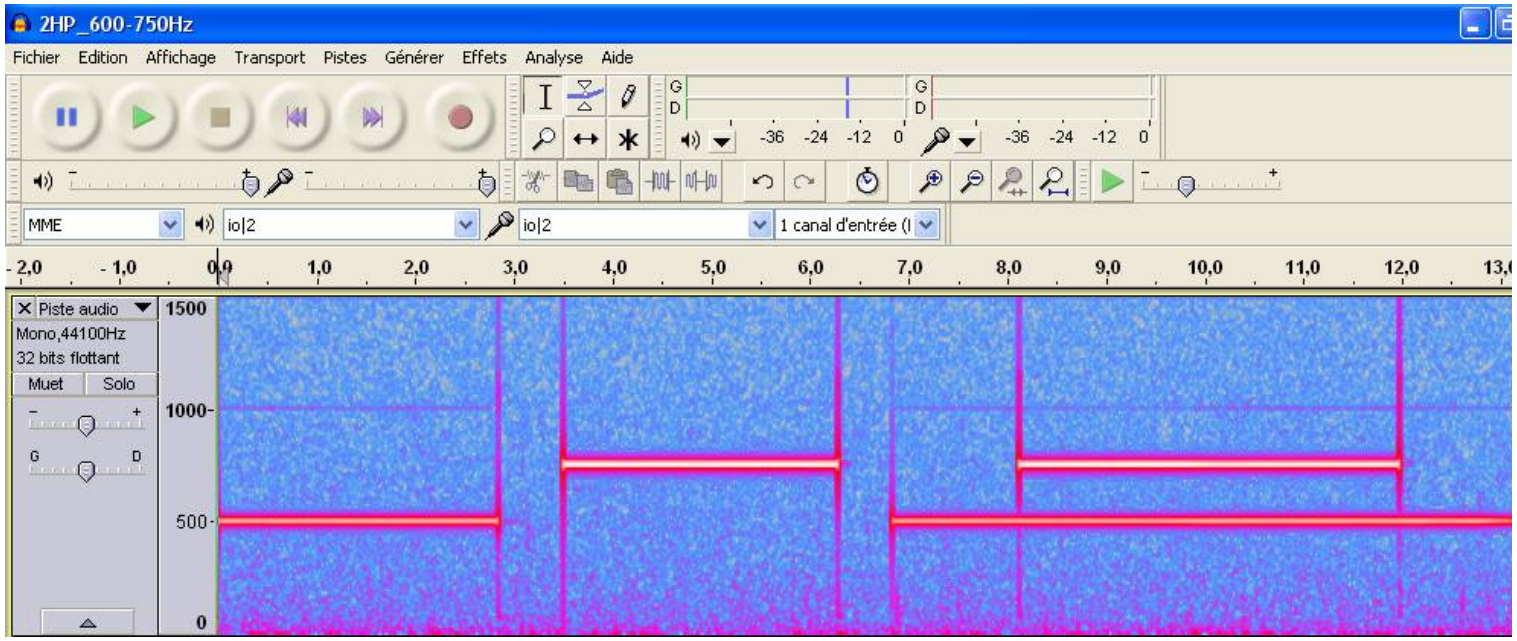




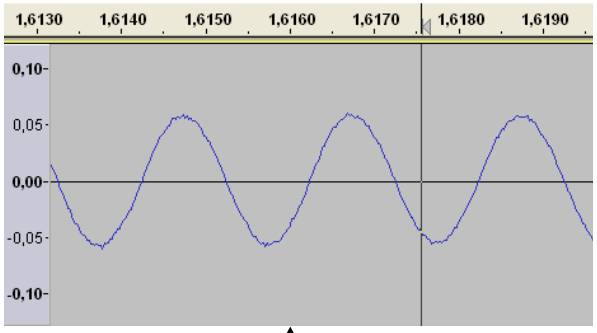
Écoutons les sons provenant de deux haut-parleurs émettant des signaux « purs » sinusoïdaux :



Spectres

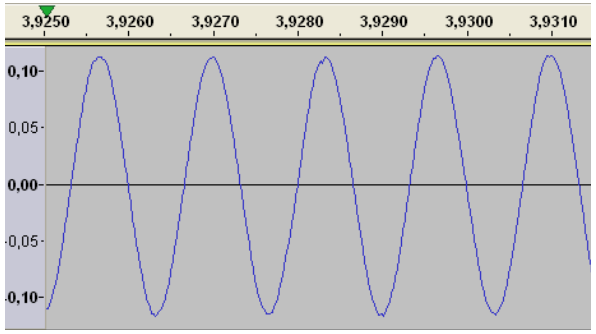


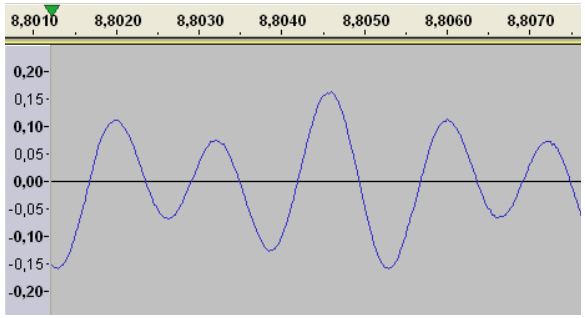
Signaux $s_1(t)$, $s_2(t)$ et $s_3(t)$



Signal 1
$f_1 = 600Hz$

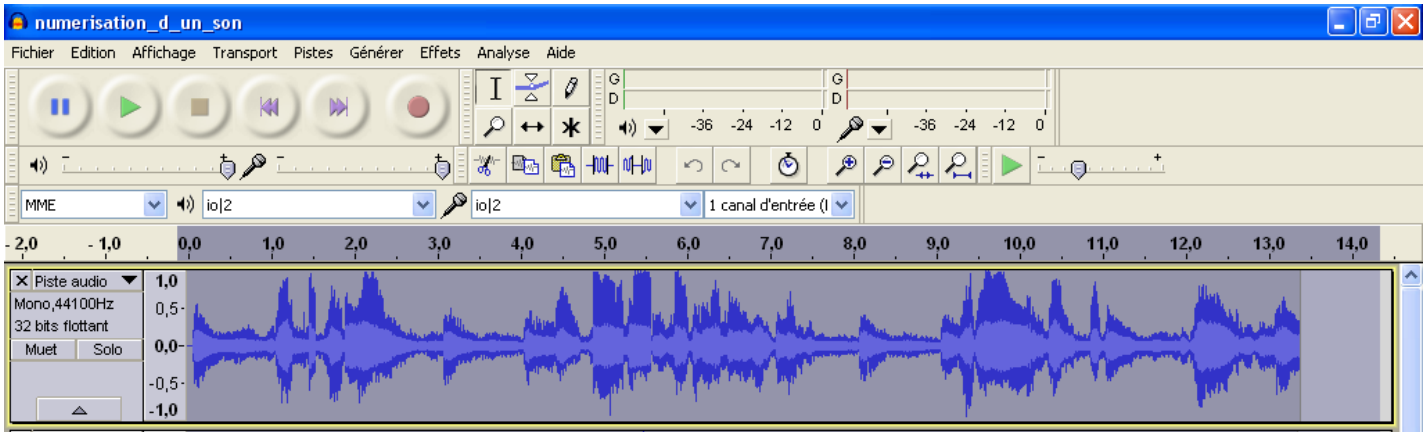
Signal 2
$f_2 = 750Hz$

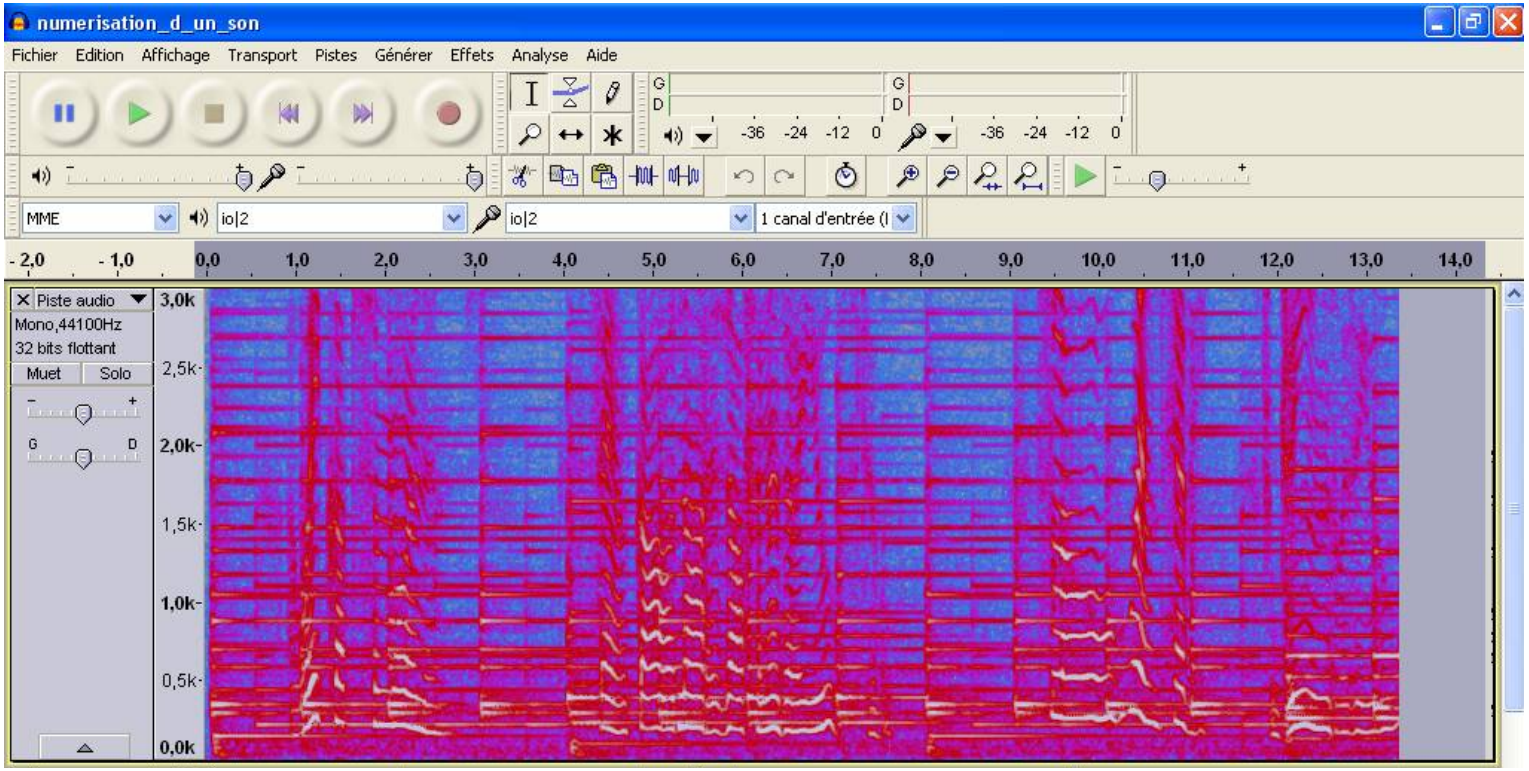




S_3 = signal 1 + signal 2
$f_1 = 600Hz$ & $f_2 = 750Hz$

Un son **musical complexe** possède un grand nombre de fréquences : les **harmoniques**.





La transformation de Fourier discrète

La transformation de Fourier Discrète s'introduit quand il s'agit de calculer la transformée de Fourier d'une fonction à l'aide d'un calculateur numérique. En effet un tel opérateur ne peut **traiter que des nombres et de plus en quantité limitée** par la taille de sa mémoire. Il s'en suit que la transformée de Fourier :

$$S(f) = \int_{-\infty}^{\infty} s(t)\, e^{-j2\pi ft}\, dt$$

doit être adaptée, d'une part en remplaçant le signal $s(t)$ par des **nombres $s(n\mathrm{T})$ qui représentent un échantillonnage** de ce signal et d'autre part en limitant l'ensemble des nombres sur lesquels portent les calculs à une valeur finie N. Le calcul fournit alors des nombres $S^*(f)$ définis

$$S^*(f) = \sum_{n=0}^{N-1} s(n\mathrm{T})\, e^{-j2\pi fn\mathrm{T}}$$

Comme le calculateur est limité dans sa puissance de calcul, il ne peut fournir ces résultats que pour un nombre limité de valeurs de la fréquence f, qu'il est naturel de choisir multiples d'un certain pas de fréquence Δf. Alors :

$$S^*(k\Delta f) = \sum_{n=0}^{N-1} s(n\mathrm{T})\, e^{-j2\pi nk\Delta f\mathrm{T}}$$

Transformation Cosinus Discrète : TCD-1D

$$F(u) = \sum_{x=0}^{N-1} \alpha(u) \times f(x) \times \cos\left[\frac{\pi.u}{2N}(2x+1)\right] \quad \longleftarrow \quad \textbf{Spectre de f(x)}$$

$$0 < u < N-1$$

$$u = 0 \Rightarrow \alpha(u) = \sqrt{\frac{1}{N}}$$

$$u \neq 0 \Rightarrow \alpha(u) = \sqrt{\frac{2}{N}}$$

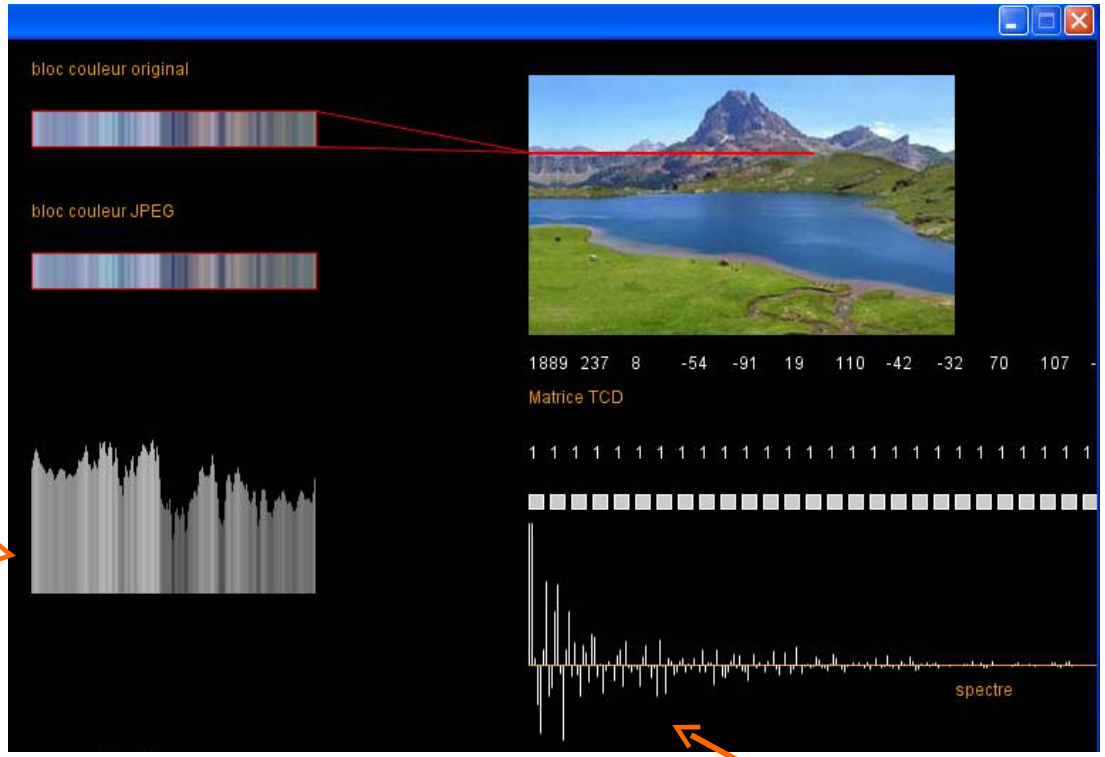
Transformation Cosinus Discrète inverse :

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) \times F(u) \times \cos\left[\frac{\pi.u}{2N}(2x+1)\right]$$

$$0 < x < N-1$$

A la recherche de fréquences dans les images

Lecture



f(x)

x, représente le n° du pixel le long d'une ligne

Spectre de f : F(u)

Ces 2 représentations sont équivalentes et on peut stocker dans la mémoire de l'ordinateur soit f(x) soit F(u).

Idée…

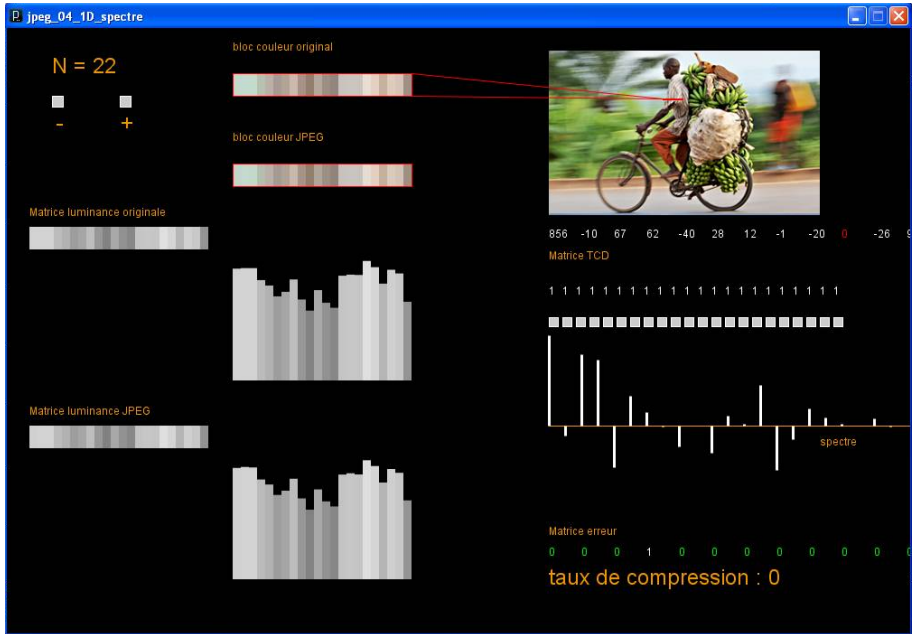
f(x) est un signal échantillonné de N valeurs.
F(u) est le spectre de f(x) et contient également N fréquences, correspondant à l'amplitude d'un cosinus.

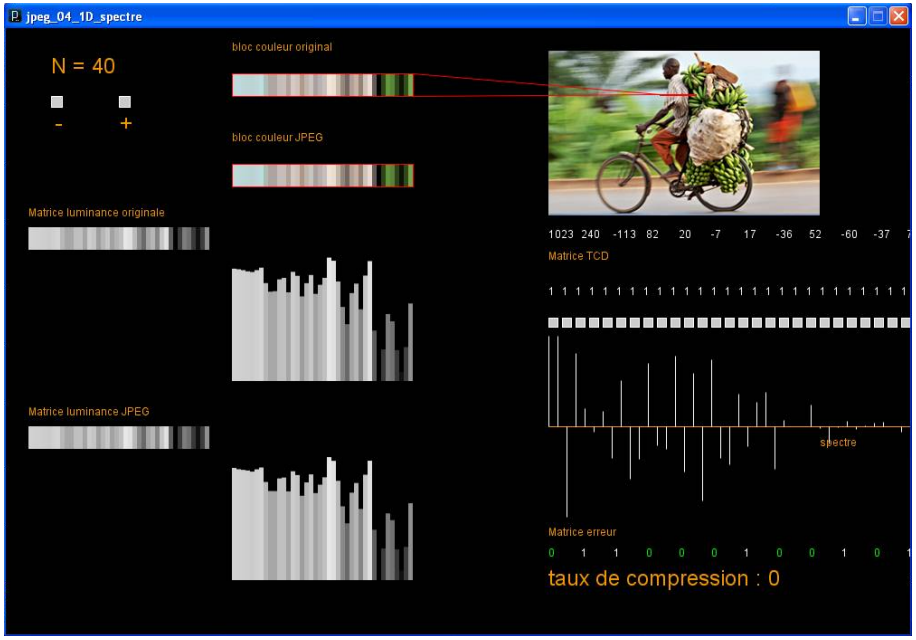
$$f(x) = \sum_{u=0}^{N-1} \alpha(u) \times F(u) \times \cos\left[\frac{\pi.u}{2N}(2x+1)\right]$$

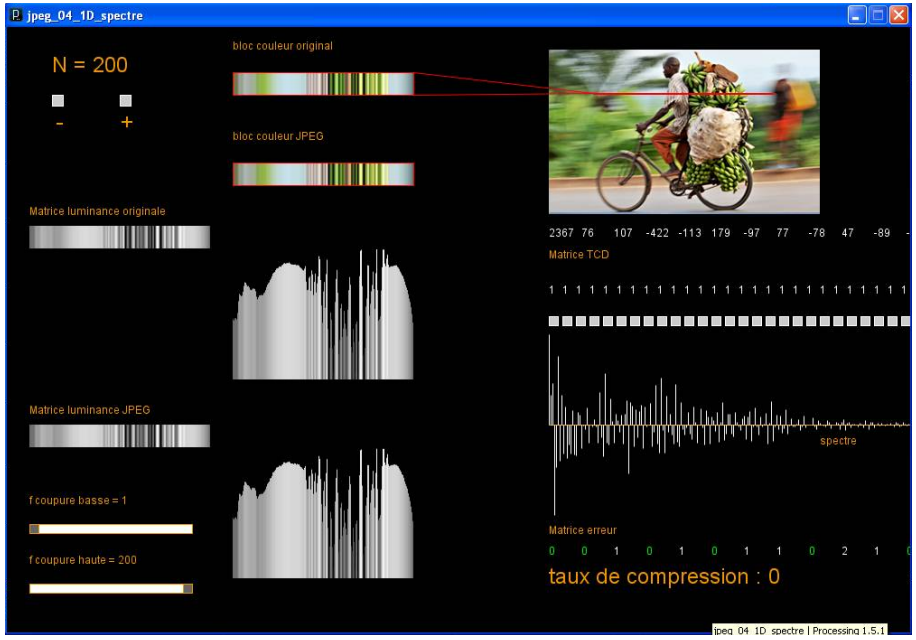
$$0 < x < N-1$$

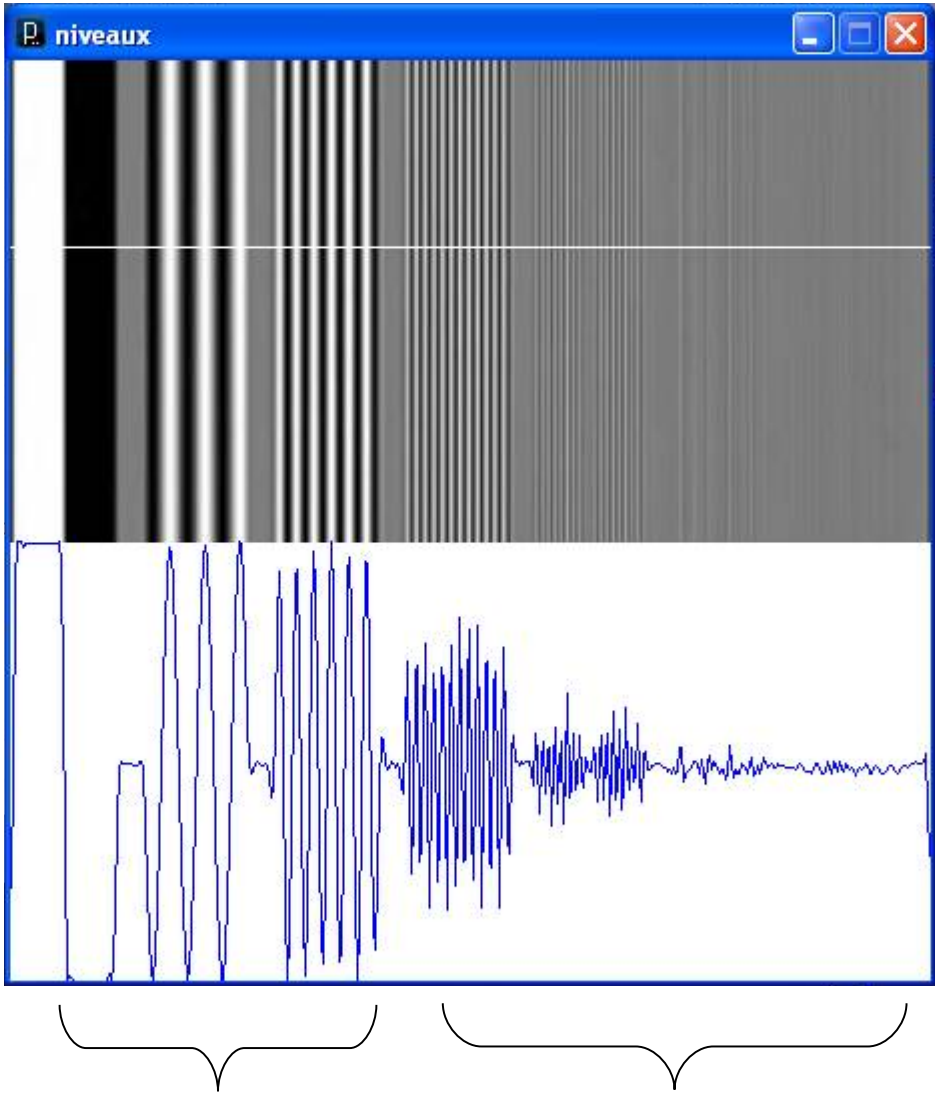
Pour gagner de la place dans la mémoire de l'ordinateur on peut envisager de ne pas stocker certaines fréquences présentes dans le spectre F(u) de f(x). On perdra bien entendu de l'information mais on gagnera de la place en mémoire. C'est le principe de la compression jpeg…


jpeg_04_1D_spectre
N = 8
bloc couleur original
bloc couleur JPEG
Matrice luminance originale
211 188 178 158 166 190 152 131
Matrice luminance JPEG
210 187 177 158 166 188 151 130
485 50 2 37 -14 0 16 -3
Matrice TCD
1 1 1 1 1 1 1 1
spectre
Matrice erreur
1 1 0 0 0 1 0 0
taux de compression : 0


jpeg_04_1D_spectre
N = 22
bloc couleur original
bloc couleur JPEG
Matrice luminance originale
Matrice luminance JPEG
856 -10 67 62 -40 28 12 -1 -20 0 -26
Matrice TCD
spectre
Matrice erreur
0 0 0 1 0 0 0 0 0 0 0 0
taux de compression : 0


jpeg_04_1D_spectre
N = 40
bloc couleur original
bloc couleur JPEG
Matrice luminance originale
Matrice luminance JPEG
1023 240 -113 82 20 -7 17 -36 52 -60 -37
Matrice TCD
spectre
Matrice erreur
0 1 1 0 0 0 1 0 0 1 0
taux de compression : 0


jpeg_04_1D_spectre
N = 200
bloc couleur original
bloc couleur JPEG
Matrice luminance originale
Matrice luminance JPEG
f coupure basse = 1
f coupure haute = 200
2367 76 107 -422 -113 179 -97 77 -78 47 -89
Matrice TCD
spectre
Matrice erreur
0 0 1 0 1 0 1 1 0 2 1
taux de compression : 0
jpeg_04_1D_spectre | Processing 1.5.1



Basses fréquences

Hautes fréquences

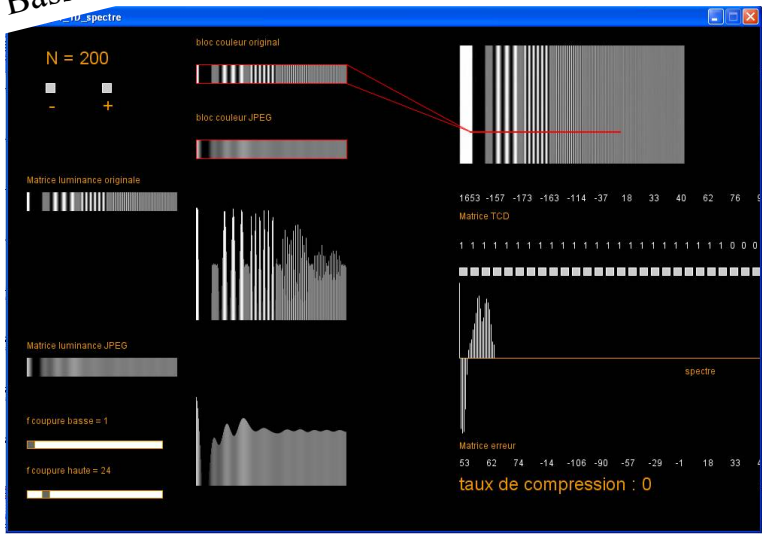
Sur cette mire on observe des bandes verticales plus ou moins serrées.

On retrouve bien ces bandes dans f(x)…

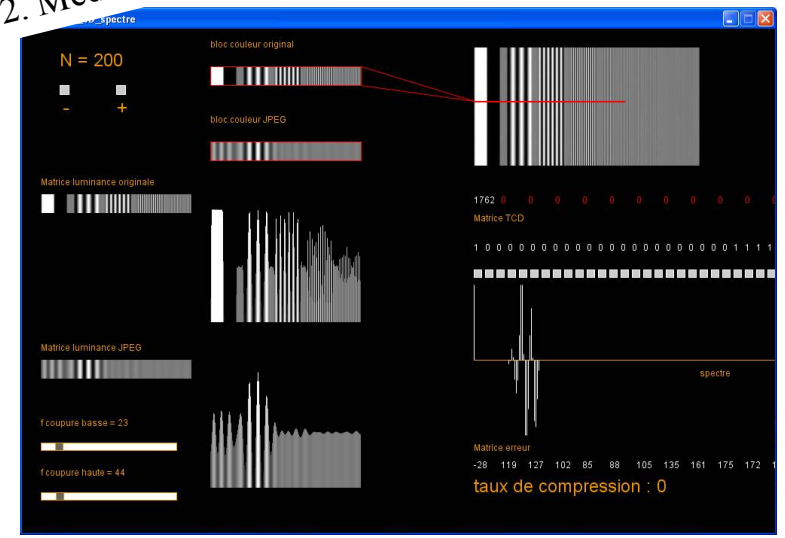
Lecture

Avec une **bande passante de 20**, on peut sélectionner certaines fréquences dans la mire et changer son aspect…

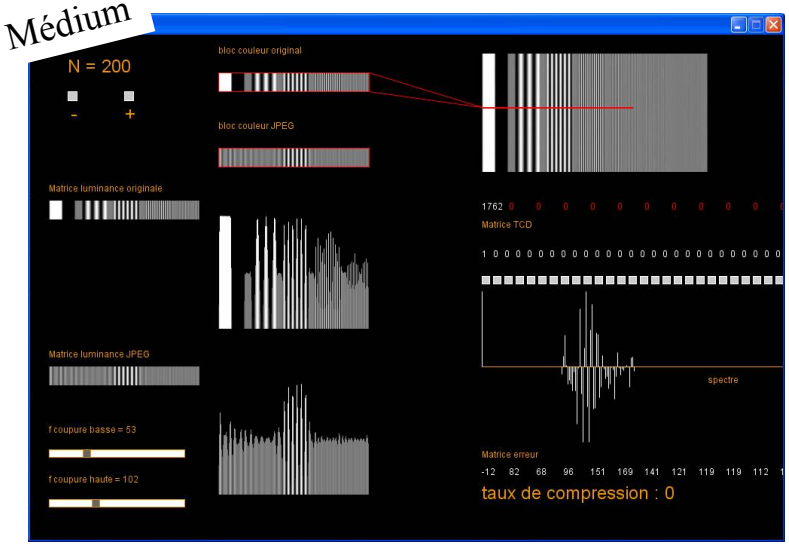
1. Basses fréquences



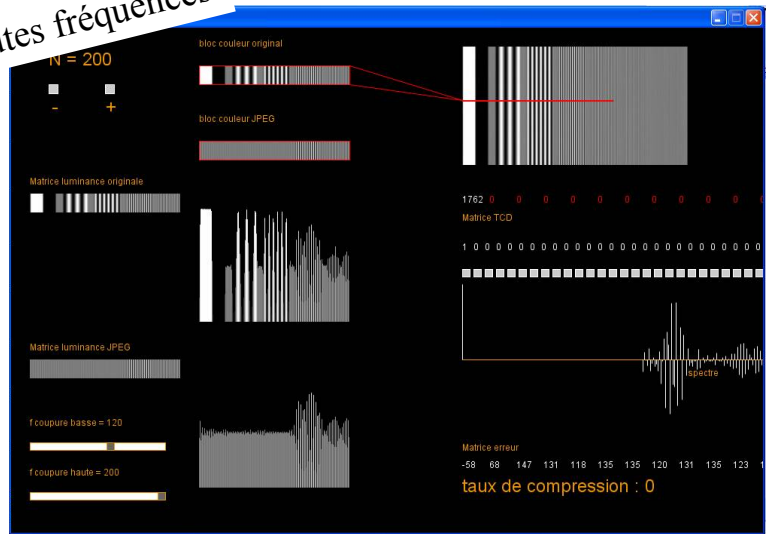
2. Médium



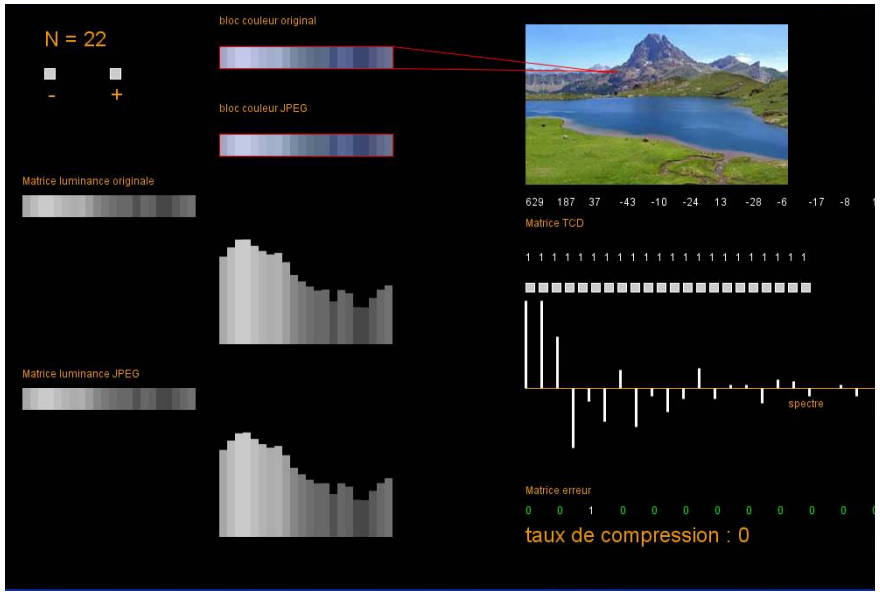
3. Médium

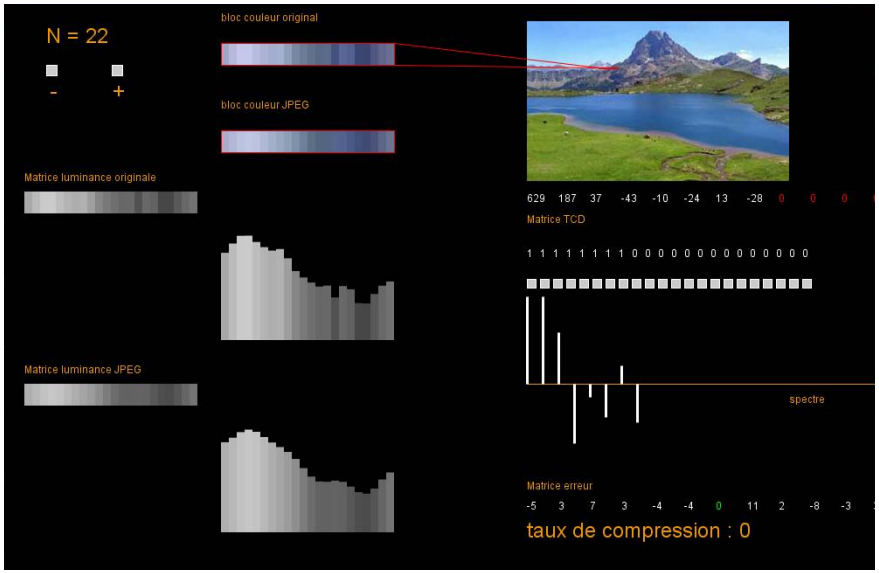


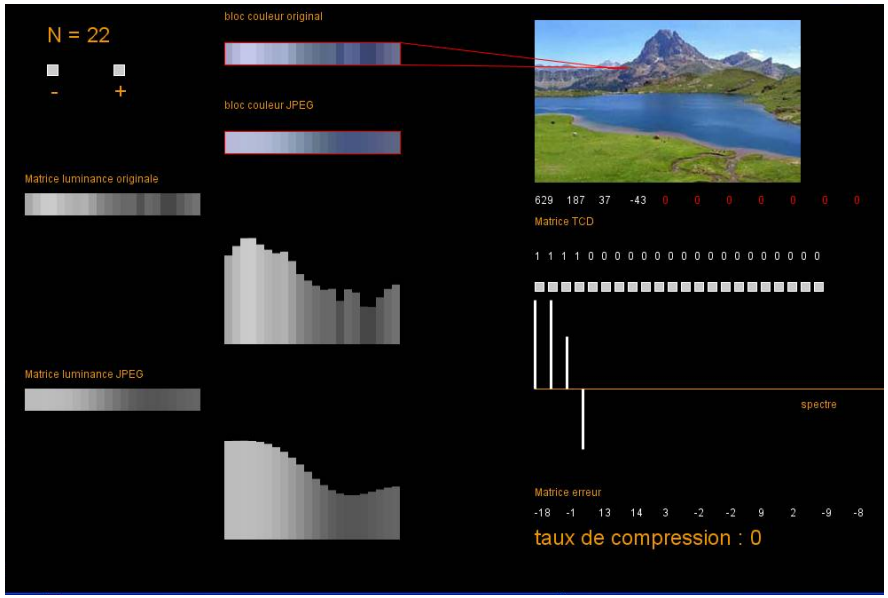
4. Hautes fréquences



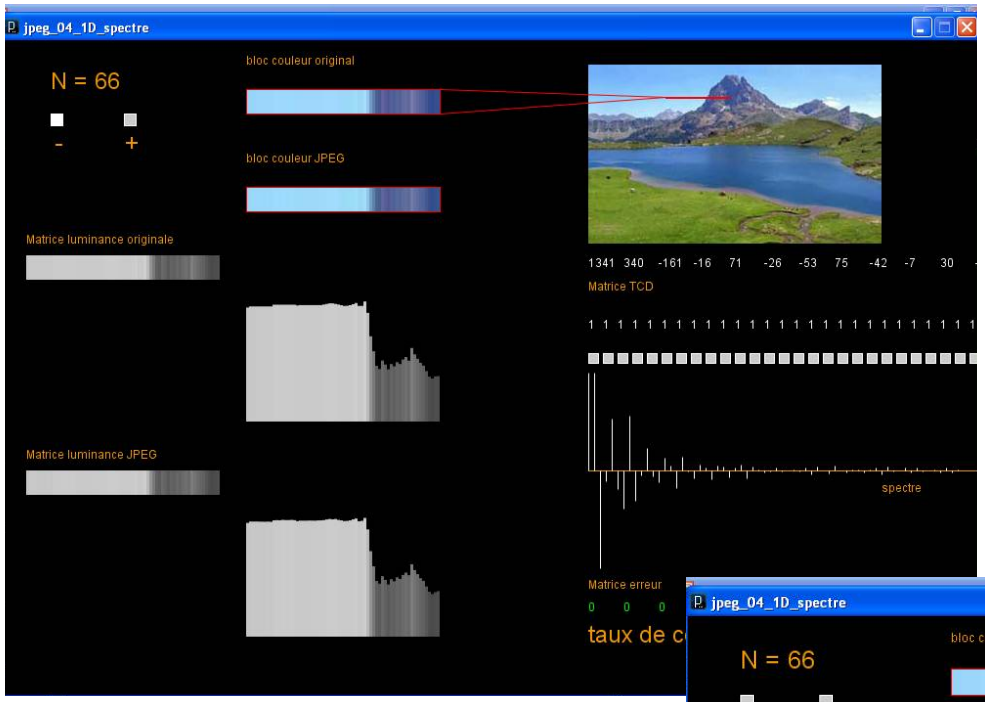
Filtrage de fréquences dans les images





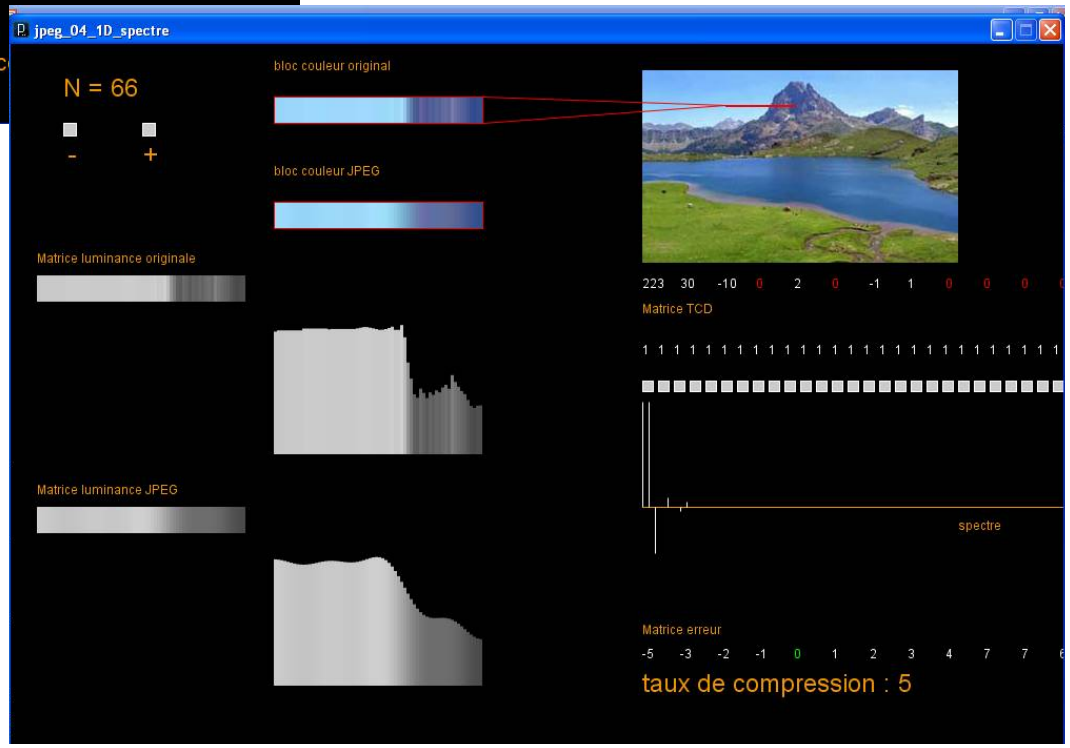


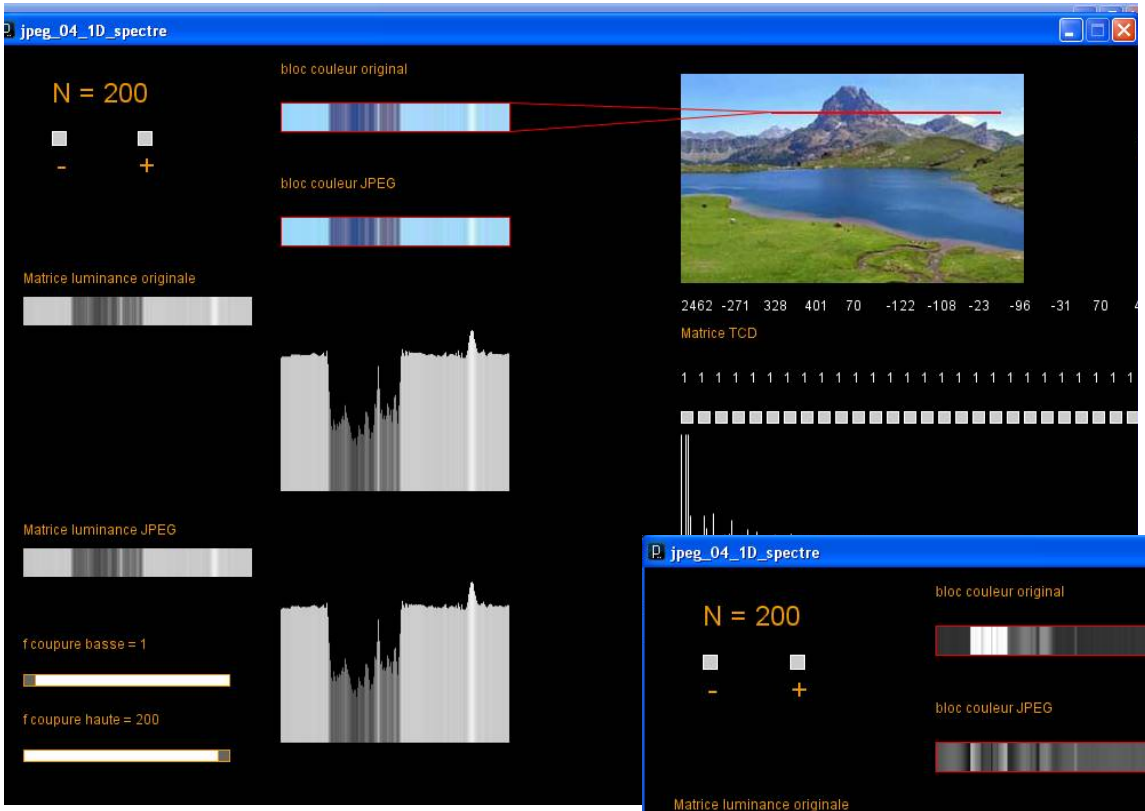
En supprimant des fréquences, on gagne de la place pour le stockage mais on perd en qualité : **compression avec pertes**. On conserve les motifs grossiers mais **on perd les détails**.



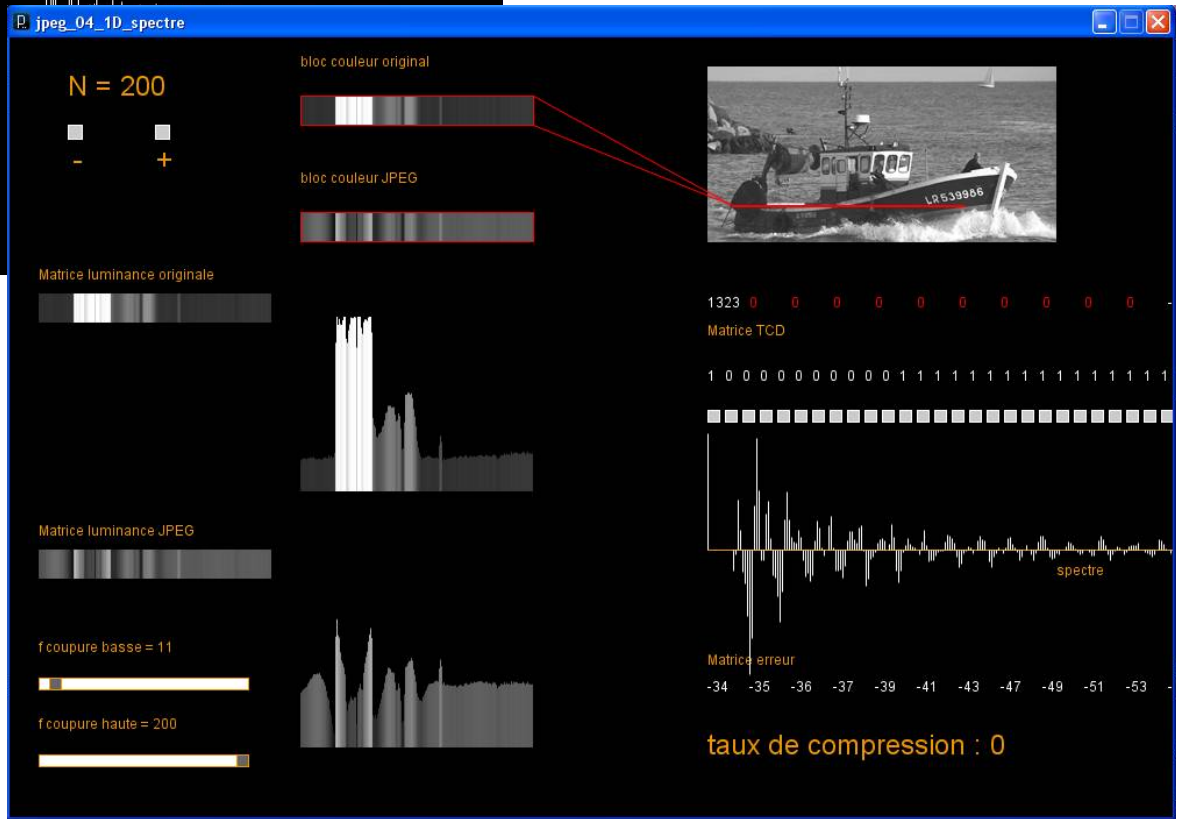
Le **taux de compression** agit sur les hautes fréquences qui sont en général d'amplitudes faibles et que l'on va encore atténuer dans le but de ne pas les stocker.

Augmentation du taux de compression



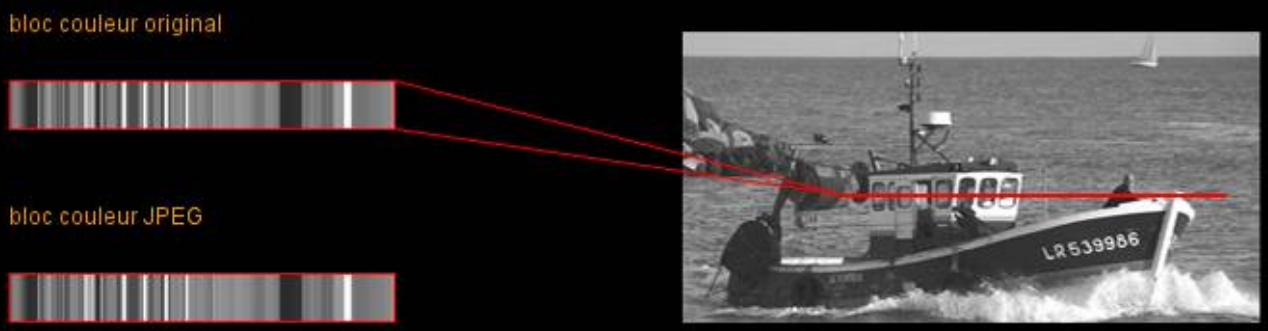


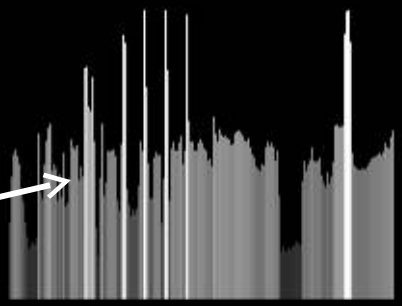
La **suppression** des **basses fréquences**…



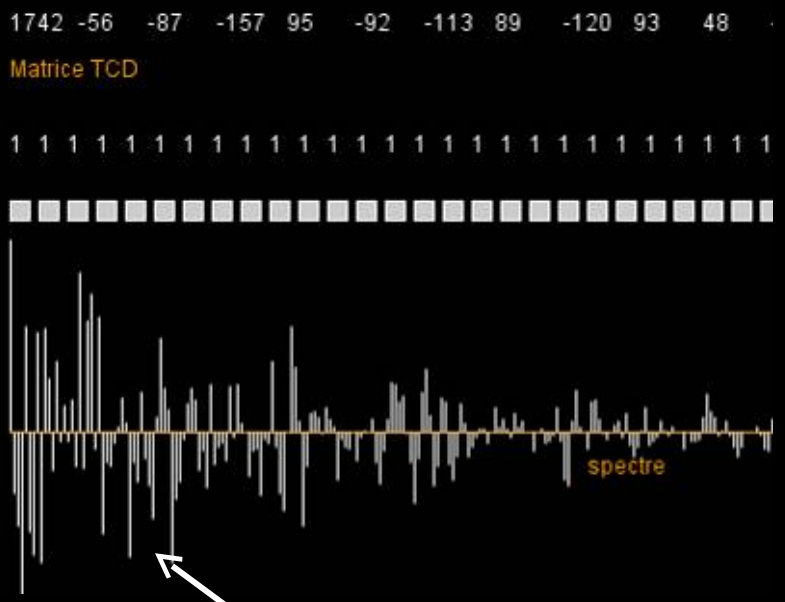
Permet la détection **des contours**.

La détection contour = filtre passe-haut





f(x)



Spectre de f : F(u)

Transformation Cosinus Discrète inverse :

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) \times F(u) \times \cos\left[\frac{\pi.u}{2N}(2x+1)\right]$$

$$0 < x < N-1$$

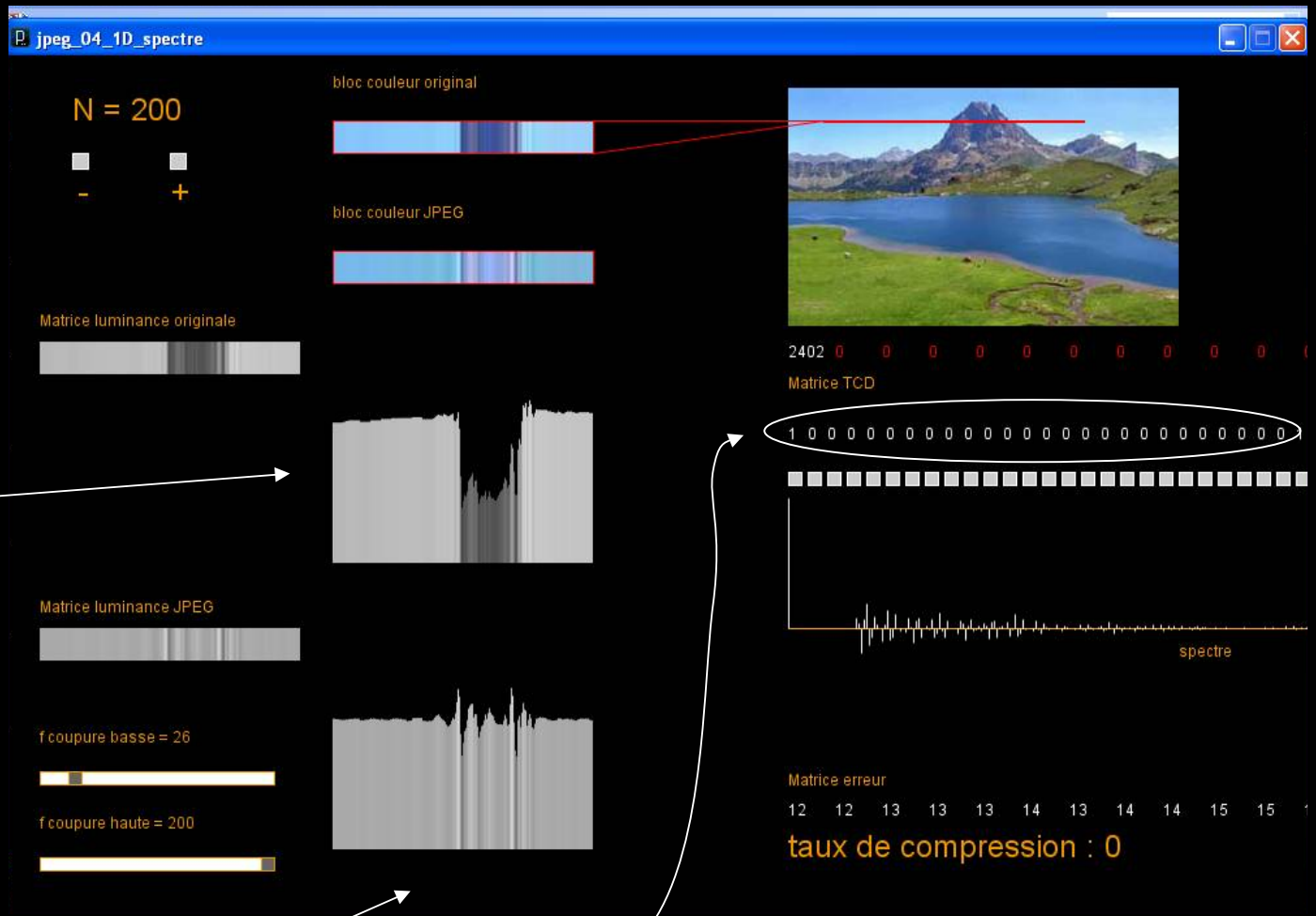
Amplitude des harmoniques

Dérivation de la TCD inverse :

$$f'(x) = \sum_{u=1}^{N-1} \alpha(u) \times F(u) \times \sin\left[\frac{\pi.u}{2N}(2x+1)\right] \times \left(-\frac{\pi.u}{N}\right)$$

$$0 < x < N-1$$

Les amplitudes des harmoniques sont multipliées par un **coefficient** qui atténue les basses fréquences. L'opération de dérivation correspond donc à un **filtrage des basses fréquences**.



Niveau de gris le long de la ligne

Niveau de gris le long de la ligne, après filtrage des basses fréquences

Conclusion

Détection de contour = « dérivation » = filtrage passe-haut

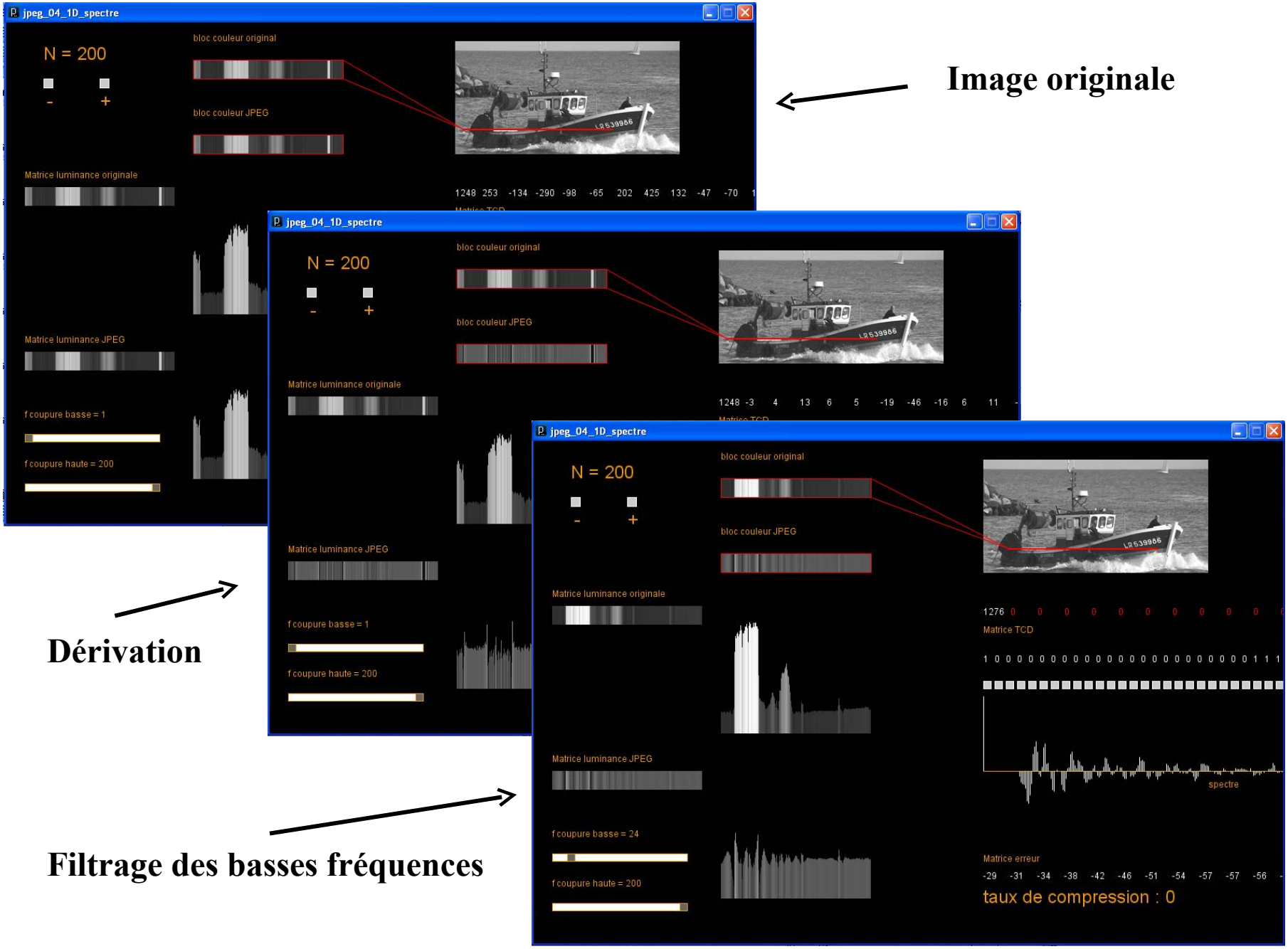
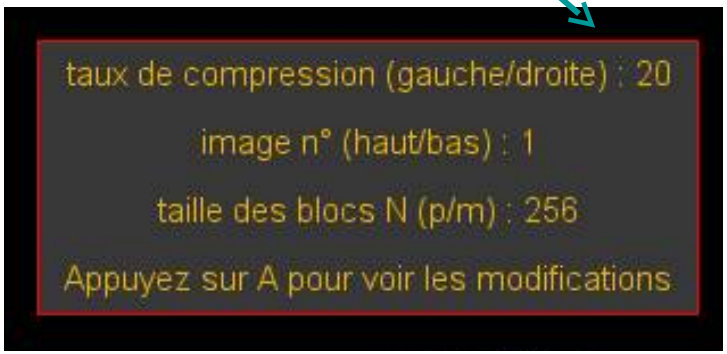


Image originale

Dérivation

Filtrage des basses fréquences

Filtrage passe-bas d'une image par compression Jpeg



F I N D E L A

D E U X I E M E P A R T I E ...