Le format de compression JPEG ou comment les mathématiques servent (parfois) à quelque chose...

E. Le Pennec
INRIA Saclay - IdF – SELECT

APMEP 24 Octobre 2010

Erwan Le Pennec

Erwan Le Pennec

 $\rightarrow$  $\rightarrow$ 

Erwan Le Pennec

 →  → 

- Bac C – lycée P. Langevin, Suresnes (93).
- Prépa – lycée Pasteur, Neuilly (93–95).
- ENS Cachan (95).
- Licence et Maîtrise de Math à Paris 7 (96).
- DEA : Mathématiques et Intelligence Artificielle – Cachan (97).
- Agrégation de Math (98).
- Doctorat : Bandelettes et représentation géométrique des images (98–02).
- Let It Wave : Traitement d'image et en particulier compression d'image de visage (02–04).
- Maître de Conférence (statistique) – Paris 7 – LPMA (04–10).
- Chargé de Recherche (statistique) – INRIA Saclay – SELECT (10–).

Compression d'image

Compression d’image

Compression d'image

$\rightarrow$ 0110101...

Compression d'image

$\rightarrow$ 0110101... $\rightarrow$

Compression d'image

Compression d'image

- Motivations ?

Compression d'image

- Motivations ?
- Compression sans perte :
  - Théorie de l'information : entropie et codage,
  - GIF, PNG...

Compression d'image

- Motivations ?
- Compression sans perte :
  - Théorie de l'information : entropie et codage,
  - GIF, PNG...
- Compression avec perte :
  - Théorie de l'approximation : approximation, quantification,
  - JPEG, JPEG2K

Compression d'image

- Motivations ?
- Compression sans perte :
  - Théorie de l'information : entropie et codage,
  - GIF, PNG...
- Compression avec perte :
  - Théorie de l'approximation : approximation, quantification,
  - JPEG, JPEG2K
- Et après ?

Image numérique

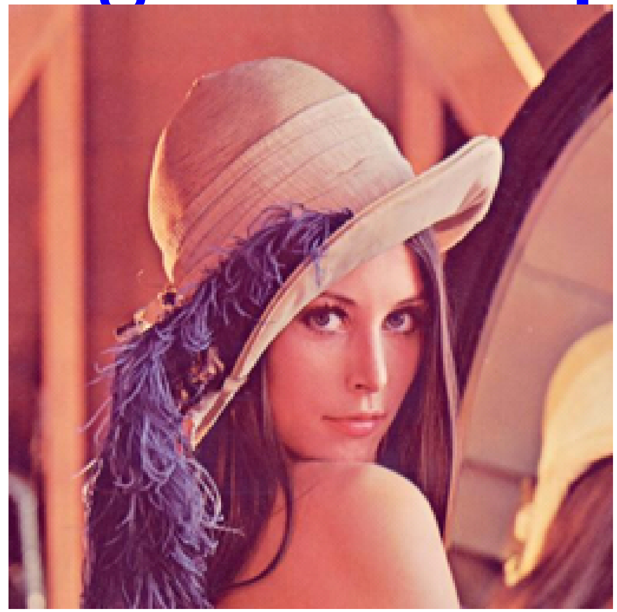
Image numérique

- Image : fonction L^2 sur un rectangle.

Image numérique

- Image : fonction L^2 sur un rectangle.
- Image numérique : version discrétisée et quantifiée ($256 = 2^8$ valeurs - 8 bits).

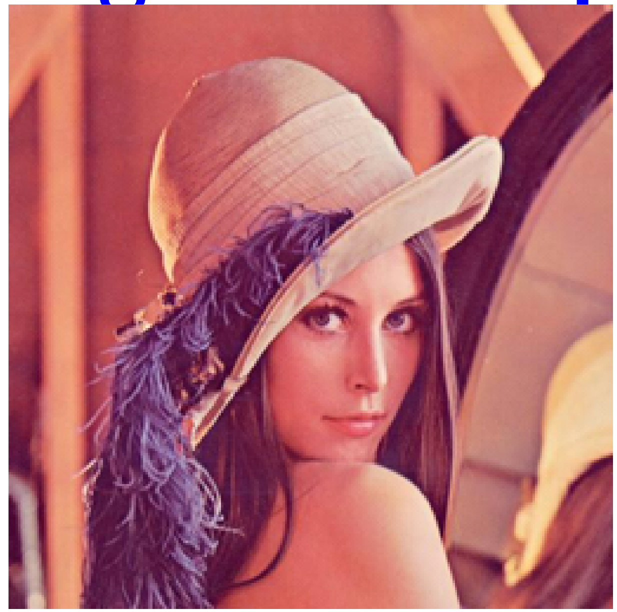
Image numérique



- Image : fonction L^2 sur un rectangle.
- Image numérique : version discrétisée et quantifiée ($256 = 2^8$ valeurs - 8 bits).
- Lien via la numérisation : c_{k_1,k_2} est une moyenne locale de f.

$$c_{k_1,k_2} = \int f(x_1,x_2)\theta(k_1\epsilon - x_1, k_2\epsilon - x_2)\mathrm{d}x_1\mathrm{d}x_2$$

Image numérique

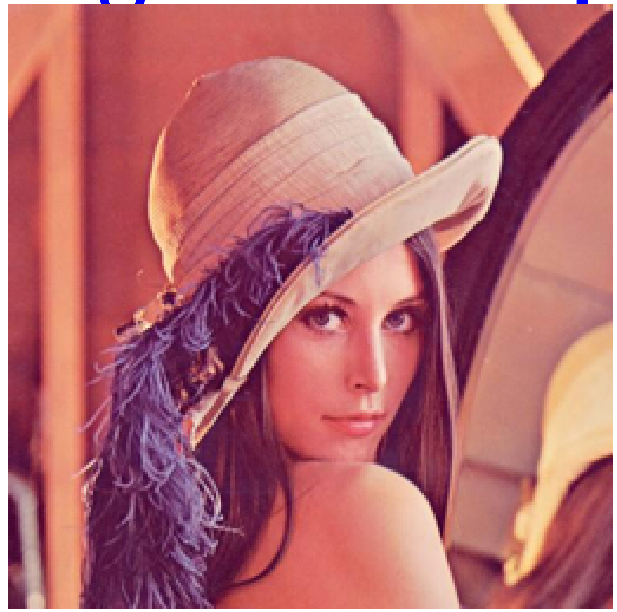


- Image : fonction L^2 sur un rectangle.
- Image numérique : version discrétisée et quantifiée ($256 = 2^8$ valeurs - 8 bits).
- Lien via la numérisation : c_{k_1,k_2} est une moyenne locale de f.

$$c_{k_1,k_2} = \int f(x_1, x_2)\theta(k_1\epsilon - x_1, k_2\epsilon - x_2)\mathrm{d}x_1\mathrm{d}x_2$$

- Pour la couleur : utilisation de 3 images (RGB - 24 bits).

Image numérique

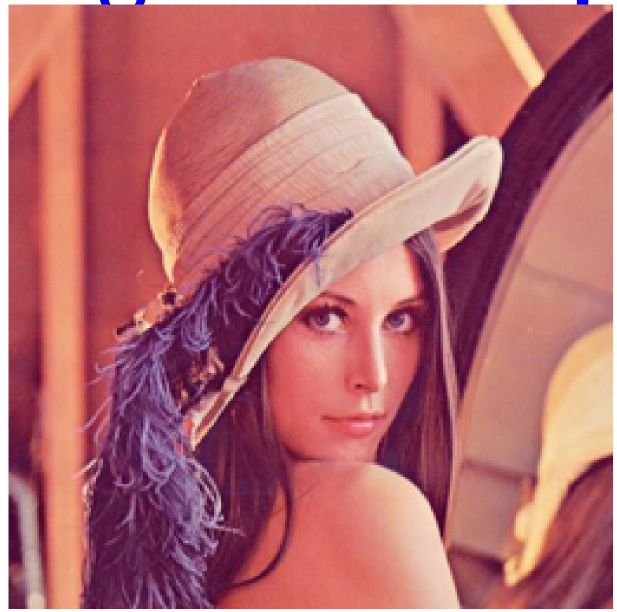


- Image : fonction L^2 sur un rectangle.
- Image numérique : version discrétisée et quantifiée ($256 = 2^8$ valeurs - 8 bits).
- Lien via la numérisation : c_{k_1,k_2} est une moyenne locale de f.

$$c_{k_1,k_2} = \int f(x_1,x_2)\theta(k_1\epsilon - x_1, k_2\epsilon - x_2)\mathrm{d}x_1\mathrm{d}x_2$$

- Pour la couleur : utilisation de 3 images (RGB - 24 bits).
- Occupation mémoire importante :
  $2048 \times 3072 \times 24 = 18 \times 2^{23}$ bits $= 18$ Mo

Image numérique



- Image : fonction L^2 sur un rectangle.
- Image numérique : version discrétisée et quantifiée ($256 = 2^8$ valeurs - 8 bits).
- Lien via la numérisation : c_{k_1,k_2} est une moyenne locale de f.

$$c_{k_1,k_2} = \int f(x_1, x_2)\theta(k_1\epsilon - x_1, k_2\epsilon - x_2)\mathrm{d}x_1\mathrm{d}x_2$$

- Pour la couleur : utilisation de 3 images (RGB - 24 bits).
- Occupation mémoire importante :
  $2048 \times 3072 \times 24 = 18 \times 2^{23}$ bits $= 18$ Mo
- Besoin de compression.

Compression d’image

Compression d'image

- Image idéale...

Compression d'image

 $\rightarrow$ 

- Image idéale...
- Image = Image numérique en niveau de gris provenant des capteurs de l'appareil photo.

Compression d'image

$\longrightarrow$

- Image idéale...
- Image = Image numérique en niveau de gris provenant des capteurs de l'appareil photo.
- Simplification ! : résolution, couleur, motif de Bayer,. . .

Compression d'image

 $\rightarrow$  $\rightleftarrows$ 0110101. . .

- Image idéale...
- Image = Image numérique en niveau de gris provenant des capteurs de l'appareil photo.
- Simplification ! : résolution, couleur, motif de Bayer,. . .
- Compression sans perte (le retour vers la version originale est possible) : GIF, PNG,. . .

Compression d'image

 $\rightarrow$  $\rightarrow$ 0110101...

- Image idéale...
- Image = Image numérique en niveau de gris provenant des capteurs de l'appareil photo.
- Simplification ! : résolution, couleur, motif de Bayer,...
- Compression sans perte (le retour vers la version originale est possible) : GIF, PNG,...
- Compression avec perte (retour impossible) : JPEG, JPEG2K,....

Compression sans perte

Compression sans perte

$\rightleftarrows$ 0110101...

Compression sans perte

$\rightleftarrows \quad 0110101\ldots$

- Retour possible = pas de perte d'information.

Compression sans perte

 $\rightleftarrows$ 0110101...

- Retour possible = pas de perte d'information.
- Exploitation de la redondance dans les suites de valeurs.

Compression sans perte

 $\rightleftarrows$ 0110101...

- Retour possible = pas de perte d'information.
- Exploitation de la redondance dans les suites de valeurs.
- Intuition : language = les mots les plus utilisés sont les plus courts...

GIF

GIF

- Algorithme de compression sans perte introduit par Compuserve en 1987.

GIF

- Algorithme de compression sans perte introduit par Compuserve en 1987.
- Image 8 bits : palette ou niveau de gris.

GIF

12	56	78	89	78	89	56	12
123	6	7	189	78	89	56	12
12	56	78	89	78	89	56	12
123	6	7	189	78	99	56	12
12	56	23	89	78	23	32	12
2	56	90	89	78	89	56	12
33	66	78	89	96	89	56	12
12	56	78	89	78	89	56	12

- Algorithme de compression sans perte introduit par Compuserve en 1987.
- Image 8 bits : palette ou niveau de gris.
- Liste des valeurs des pixels (ligne par ligne).

GIF

12	56	78	89	78	89	56	12
123	6	7	189	78	89	56	12
12	56	78	89	78	89	56	12
123	6	7	189	78	99	56	12
12	56	23	89	78	23	32	12
2	56	90	89	78	89	56	12
33	66	78	89	96	89	56	12
12	56	78	89	78	89	56	12

$\downarrow$ liste

12	56	78	89	78	89	56	12	123	6	7	189	78	89	56	12	12	56	78	89	78	89	56	12	123	6	7	189	78	99	56	12

12	56	23	89	78	23	32	12	2	56	90	89	78	89	56	12	33	66	78	89	96	89	56	12	12	56	78	89	78	89	56	12

- Algorithme de compression sans perte introduit par Compuserve en 1987.
- Image 8 bits : palette ou niveau de gris.
- Liste des valeurs des pixels (ligne par ligne).
- Répétition dans la liste .

GIF

12	56	78	89	78	89	56	12
123	6	7	189	78	89	56	12
12	56	78	89	78	89	56	12
123	6	7	189	78	99	56	12
12	56	23	89	78	23	32	12
2	56	90	89	78	89	56	12
33	66	78	89	96	89	56	12
12	56	78	89	78	89	56	12

$\downarrow$ liste

12	56	78	89	78	89	56	12	123	6	7	189	78	89	56	12	12	56	78	89	78	89	56	12	123	6	7	189	78	99	56	12

12	56	23	89	78	23	32	12	2	56	90	89	78	89	56	12	33	66	78	89	96	89	56	12	12	56	78	89	78	89	56	12

- Algorithme de compression sans perte introduit par Compuserve en 1987.
- Image 8 bits : palette ou niveau de gris.
- Liste des valeurs des pixels (ligne par ligne).
- Répétition dans la liste .
- Comment exploiter cette structure ? : compression.

GIF

12	56	78	89	78	89	56	12
123	6	7	189	78	89	56	12
12	56	78	89	78	89	56	12
123	6	7	189	78	99	56	12
12	56	23	89	78	23	32	12
2	56	90	89	78	89	56	12
33	66	78	89	96	89	56	12
12	56	78	89	78	89	56	12

$\downarrow$ liste

12	56	78	89	78	89	56	12	123	6	7	189	78	89	56	12	12	56	78	89	78	89	56	12	123	6	7	189	78	99	56	12

12	56	23	89	78	23	32	12	2	56	90	89	78	89	56	12	33	66	78	89	96	89	56	12	12	56	78	89	78	89	56	12

- Algorithme de compression sans perte introduit par Compuserve en 1987.
- Image 8 bits : palette ou niveau de gris.
- Liste des valeurs des pixels (ligne par ligne).
- Répétition dans la liste .
- Comment exploiter cette structure ? : compression.
- Exemple simple : algorithme de type Run Length Encoding qui exploite les suites de valeurs identiques...

Compression

Compression

- Symboles x avec $x \in \mathcal{X}$

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \ldots x_n \in \mathcal{X}^n$

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

$$x_1 \dots x_n \to C(x_1 \dots x_n)$$

avec $C(x_1 \dots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \dots x_n)$.

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

$$x_1 \dots x_n \to C(x_1 \dots x_n)$$

  avec $C(x_1 \dots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \dots x_n)$.
- Codage le plus simple :

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

$$x_1 \dots x_n \to C(x_1 \dots x_n)$$

avec $C(x_1 \dots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \dots x_n)$.

- Codage le plus simple :
  - énumération des suites possibles : $(x_1 \dots x_n) \in \{0,1\}^{n \log_2(|\mathcal{X}|)}$,

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

$$x_1 \dots x_n \to C(x_1 \dots x_n)$$

avec $C(x_1 \dots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \dots x_n)$.

- Codage le plus simple :
  - énumération des suites possibles : $(x_1 \dots x_n) \in \{0, 1\}^{n \log_2(|\mathcal{X}|)}$,
  - suites codées par le son numéro dans la liste en binaire.

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \dots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

$$x_1 \dots x_n \to C(x_1 \dots x_n)$$

  avec $C(x_1 \dots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \dots x_n)$.
- Codage le plus simple :
  - énumération des suites possibles : $(x_1 \dots x_n) \in \{0, 1\}^{n \log_2(|\mathcal{X}|)}$,
  - suites codées par le son numéro dans la liste en binaire.
- Limitation (lemme des tiroirs) : impossible de comprimer toutes les séquences...,

Compression

- Symboles x avec $x \in \mathcal{X}$ (ici valeur du pixel entre 0 et 255).
- Suite de n symboles $x_1 \ldots x_n \in \mathcal{X}^n$ (ici listes des valeurs de pixels dans l'image)
- Codage sans perte : application injective

  $$x_1 \ldots x_n \rightarrow C(x_1 \ldots x_n)$$

  avec $C(x_1 \ldots x_n)$ une suite de 0 et de 1 de longueur $l(x_1 \ldots x_n)$.
- Codage le plus simple :
  - énumération des suites possibles : $(x_1 \ldots x_n) \in \{0, 1\}^{n \log_2(|\mathcal{X}|)}$,
  - suites codées par le son numéro dans la liste en binaire.
- Limitation (lemme des tiroirs) : impossible de comprimer toutes les séquences...,
- Codage plus court en moyenne $\rightarrow$ modèle statistique de la complexité des suites (images).

Entropie

Entropie

- Comment “mesurer” la complexité d’une séquence ?

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.
- Séquence complexe $\simeq$ séquence rare.

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.
- Séquence complexe $\simeq$ séquence rare.
- Bon concept : entropie statistique (Shannon).

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.
- Séquence complexe $\simeq$ séquence rare.
- Bon concept : entropie statistique (Shannon).

$$H = -\sum_{x \in \mathcal{X}} P_x \log_2 P_x$$

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.
- Séquence complexe $\simeq$ séquence rare.
- Bon concept : entropie statistique (Shannon).

$$H = -\sum_{x \in \mathcal{X}} P_x \log_2 P_x$$

- Exemples :
  - $P_x = 1/(|\mathcal{X}|^n) \to H = n \log_2(|\mathcal{X}|)$,
  - Si $\exists x_0, P_{x_0} = 1$ alors $H = 0$.

Entropie

- Comment “mesurer” la complexité d’une séquence ?
- Modèle probabiliste sur les séquences : $P(x) = P_x$.
- Séquence complexe $\simeq$ séquence rare.
- Bon concept : entropie statistique (Shannon).

$$H = -\sum_{x \in \mathcal{X}} P_x \log_2 P_x$$

- Exemples :
  - $P_x = 1/(|\mathcal{X}|^n) \rightarrow H = n \log_2(|\mathcal{X}|)$,
  - Si $\exists x_0, P_{x_0} = 1$ alors $H = 0$.
- Propriété importante : $0 \leq H(X) \leq \log_2 |\mathcal{X}|$

Principe d'Équirépartition Asymptotique

Principe d'Équirépartition Asymptotique

- Asymptotiquement « presques toutes « les suites se comportent de la même manière vis à vis de l'entropie !

Principe d'Équirépartition Asymptotique

- Asymptotiquement « presques toutes « les suites se comportent de la même manière vis à vis de l'entropie !
- $X_1, \ldots, X_n$ i.i.d. $\sim P$.

Principe d'Équirépartition Asymptotique

- Asymptotiquement « presques toutes « les suites se comportent de la même manière vis à vis de l'entropie !
- $X_1, \ldots, X_n$ i.i.d. $\sim P$.
- LGN : $-\frac{1}{n}\sum_i \log_2(P_{X_i}) \to H(X) = E_P(-\log_2 P_X)$.

Principe d'Équirépartition Asymptotique

- Asymptotiquement « presques toutes « les suites se comportent de la même manière vis à vis de l'entropie !
- $X_1, \ldots, X_n$ i.i.d. $\sim P$.
- LGN : $-\frac{1}{n} \sum_i \log_2(P_{X_i}) \to H(X) = E_P(-\log_2 P_X)$.
- $A_n^\epsilon = \{x_1 \cdots x_n, | -\frac{1}{n}, \sum_i \log_2(P_{X_i}) - H(X)| < \epsilon\}$.

Principe d'Équirépartition Asymptotique

- Asymptotiquement « presques toutes « les suites se comportent de la même manière vis à vis de l'entropie !
- $X_1, \ldots, X_n$ i.i.d. $\sim P$.
- LGN : $-\frac{1}{n}\sum_i \log_2(P_{X_i}) \to H(X) = E_P(-\log_2 P_X)$.
- $A_n^\epsilon = \{x_1 \cdots x_n, |-\frac{1}{n}, \sum_i \log_2(P_{X_i}) - H(X)| < \epsilon\}$.
- Prop :
  - $A_n^\epsilon = \{x_1 \cdots x_n, 2^{-n(H+\epsilon)} \leq P_{x_1 \cdots x_n} \leq 2^{-n(H-\epsilon)}\}$
  - $|A_n^\epsilon| \leq 2^{n(H+\epsilon)}$
  - $P\{A_n^\epsilon\} \geq 1 - \epsilon$ pour n assez grand
  - $|A_n^\epsilon| \geq (1-\epsilon)2^{n(H-\epsilon)}$ pour n assez grand

Codage et AEP

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.
- Coût par mot : $n \log_2 |\mathcal{X}|$ bits.

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.
- Coût par mot : $n \log_2 |\mathcal{X}|$ bits.
- Coût par symbole : $\log_2 |\mathcal{X}|$ bits.

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.
- Coût par mot : $n \log_2 |\mathcal{X}|$ bits.
- Coût par symbole : $\log_2 |\mathcal{X}|$ bits.
- Amélioration avec A_n^ϵ :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - faire une liste des $2^{nH+\epsilon}$ mots dans A_n^ϵ,
  - coder chaque mot par son appartenance (ou non) à A_n^ϵ suivi de sa position dans la liste correspondante.

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.
- Coût par mot : $n \log_2 |\mathcal{X}|$ bits.
- Coût par symbole : $\log_2 |\mathcal{X}|$ bits.
- Amélioration avec A_n^ϵ :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - faire une liste des $2^{nH+\epsilon}$ mots dans A_n^ϵ,
  - coder chaque mot par son appartenance (ou non) à A_n^ϵ suivi de sa position dans la liste correspondante.
- Coût moyen par mot :
  $(1-\epsilon)(1+n(H+\epsilon)) + \epsilon(1+n\log_2|\mathcal{X}|) \simeq n(H+\epsilon(1+\log_2|\mathcal{X}|-H))$

Codage et AEP

- Algorithme de codage le plus simple :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - coder chaque mot par sa position dans la liste.
- Coût par mot : $n \log_2 |\mathcal{X}|$ bits.
- Coût par symbole : $\log_2 |\mathcal{X}|$ bits.
- Amélioration avec A_n^ϵ :
  - faire une liste des $|\mathcal{X}|^n$ mots possibles,
  - faire une liste des $2^{nH+\epsilon}$ mots dans A_n^ϵ,
  - coder chaque mot par son appartenance (ou non) à A_n^ϵ suivi de sa position dans la liste correspondante.
- Coût moyen par mot :
  $(1-\epsilon)(1+n(H+\epsilon)) + \epsilon(1+n\log_2|\mathcal{X}|) \simeq n(H+\epsilon(1+\log_2|\mathcal{X}|-H))$
- Coût moyen par symbole $\simeq (H+\epsilon')$

Théorème de Shannon

Théorème de Shannon

- On code $x_1 \cdots x_n$ par $C(x_1 \cdots x_n)$ de longueur $l(x_1 \cdots x_n)$.

Théorème de Shannon

- On code $x_1 \cdots x_n$ par $C(x_1 \cdots x_n)$ de longueur $l(x_1 \cdots x_n)$.
- Restriction sur les codes possibles :
  - Injectivité (non singulier indispensable) :
    $x_1 \cdots x_n \neq x'_1 \cdots x'_n \implies C(x_1 \cdots x_n) \neq C(x'_1 \cdots x'_n)$,
  - Code par extension (pratique) : $C(x_1 \cdots x_n) = C(x_1) \cdots C(x_n)$.

Théorème de Shannon

- On code $x_1 \cdots x_n$ par $C(x_1 \cdots x_n)$ de longueur $l(x_1 \cdots x_n)$.
- Restriction sur les codes possibles :
  - Injectivité (non singulier indispensable) : $x_1 \cdots x_n \neq x_1' \cdots x_n' \implies C(x_1 \cdots x_n) \neq C(x_1' \cdots x_n')$,
  - Code par extension (pratique) : $C(x_1 \cdots x_n) = C(x_1) \cdots C(x_n)$.
- Th : pour tout code par extension injectif (uniquement décodable)

$$E(l_x) = \sum_x P_x l_x \geq H(X)$$

Théorème de Shannon

- On code $x_1 \cdots x_n$ par $C(x_1 \cdots x_n)$ de longueur $l(x_1 \cdots x_n)$.
- Restriction sur les codes possibles :
  - Injectivité (non singulier indispensable) :
    $x_1 \cdots x_n \neq x'_1 \cdots x'_n \implies C(x_1 \cdots x_n) \neq C(x'_1 \cdots x'_n)$,
  - Code par extension (pratique) : $C(x_1 \cdots x_n) = C(x_1) \cdots C(x_n)$.
- Th : pour tout code par extension injectif (uniquement décodable)

$$E(l_x) = \sum_x P_x l_x \geq H(X)$$

- On ne peut pas faire (beaucoup) mieux que le code naïf précédent !

Théorème de Shannon

- On code $x_1 \cdots x_n$ par $C(x_1 \cdots x_n)$ de longueur $l(x_1 \cdots x_n)$.
- Restriction sur les codes possibles :
  - Injectivité (non singulier indispensable) :
    $x_1 \cdots x_n \neq x'_1 \cdots x'_n \implies C(x_1 \cdots x_n) \neq C(x'_1 \cdots x'_n)$,
  - Code par extension (pratique) : $C(x_1 \cdots x_n) = C(x_1) \cdots C(x_n)$.
- Th : pour tout code par extension injectif (uniquement décodable)

$$E(l_x) = \sum_x P_x l_x \geq H(X)$$

- On ne peut pas faire (beaucoup) mieux que le code naïf précédent !
- Preuve :. . . .

Kraft/Mc Millan

- $\mathsf{Th} : \sum_{x \in \mathcal{X}} 2^{-l_x} \leq 1.$

Kraft/Mc Millan

- Th : $\sum_{x \in \mathcal{X}} 2^{-l_x} \leq 1$.
- Preuve : On code x par c_x de longueur l_x et donc $\forall n$, $x_{i_1} \ldots, x_{i_n}$ par $c_{x_{i_1}} \cdots c_{x_{i_n}}$ de longueur $l_{x_{i_1}} + \cdots + c_{x_{i_n}}$.
  On suppose $\mathcal{X}$ fini et on note $l_{\max}$ le mot le plus long.
  On suppose que le code est admissible : les séquences de tailles n s'injectent dans les mots de tailles inférieur à $nl_{\max}$.

  $$\sum_{(x_1,\ldots,x_n)\in\mathcal{X}^n} 2^{-(l_{x_{i_1}}+\cdots+l_{x_{i_n}})} \leq \sum_{l \leq nl_{\max}} 2^l 2^{-l} = nl_{\max}$$

  et donc

  $$\sum_{x\in\mathcal{X}} 2^{-l_x} \leq (nl_{\max})^{1/n} \xrightarrow[n\to\infty]{} 1$$

  Si $\mathcal{X}$ est dénombrable, on obtient le même résultat en passant à la limite sur des sous-ensembles finis croissants de $\mathcal{X}$.

Shannon

Shannon

- $\mathrm{Th}: \sum_{x \in \mathcal{X}} P_x l_x \geq -\sum_{x \in \mathcal{X}} P_x \log_2 P_x = H$

Shannon

- Th : $\sum_{x\in\mathcal{X}} P_x l_x \geq -\sum_{x\in\mathcal{X}} P_x \log_2 P_x = H$
- Preuve

  Longueur moyenne du code : $\sum_x P_x l_x$

  Contrainte sur les codes (Kraft) : $K = \sum_x 2^{-l_x} \leq 1$.

  On pose $Q_x = 2^{-l_x}/K$:

$$\sum_x P_x l_x + \sum_x P_x \log_2 P_i \sum_x P_x \log_2 (K/Q_x) + \sum_x P_x \log_2 P_i$$

$$= \sum_x P_x \log_2 \left(\frac{P_x}{Q_x}\right) - \log_2 K$$

$$= D(P|Q) - \log_2 K \geq 0$$

où D est la divergence de Küllback-Leibler (égalité ssi $l_x = -\log_2 P_x$.)

Rappel :

$$D(P|Q) = \sum_x P_x \log_2 \left(\frac{P_x}{Q_x}\right) = \sum_x P_x - \log_2 \left(\frac{Q_x}{P_x}\right)$$

Par convexité de $-\log_2 x$ et Jensen

$$\geq -\log_2 \left(\sum_x P_x \frac{Q_x}{P_x}\right) = -\log_2 \left(\sum_x Q_x\right) = 0$$

Codage entropique

Codage entropique

- Pour des P_x (distribution de probabilité) connus, problème résolu.

Codage entropique

- Pour des P_x (distribution de probabilité) connus, problème résolu.
- Algorithmes efficaces pour obtenir un codage quasi optimal (mot de longueur $\simeq -\log_2 P_x$) :
  - Huffman (arbre)
  - Codage arithmétique (plus efficace mais plus complexe)

Shannon et Huffman

Shannon et Huffman

- Codage associé à des arbres dyadiques (branches codées par 0 ou 1)

Shannon et Huffman

- Codage associé à des arbres dyadiques (branches codées par 0 ou 1)
- Codage de Shannon-Fano (stratégie de haut en bas) :
  - ordonner les x par ordre de probabilité (initialisation),
  - couper récursivement en ensembles équiprobables (création d'un arbre)
  - coder x à l'aide des valeurs des branches de haut en bas : $l_x \leq \ulcorner \log_2 P_x \urcorner + 1$.
  - $H(X) \leq E(l_x) \leq H(X) + 2$

Shannon et Huffman

- Codage associé à des arbres dyadiques (branches codées par 0 ou 1)
- Codage de Shannon-Fano (stratégie de haut en bas) :
  - ordonner les x par ordre de probabilité (initialisation),
  - couper récursivement en ensembles équiprobables (création d'un arbre)
  - coder x à l'aide des valeurs des branches de haut en bas : $l_x \leq \ulcorner \log_2 P_x \urcorner + 1$.
  - $H(X) \leq E(l_x) \leq H(X) + 2$
- Codage d'Huffman (stratégie de bas en haut) :
  - regrouper de manière récursive les deux symboles les moins probables (création d'une branche 0/1),
  - coder x à l'aide des valeurs des branches de haut en bas.
  - $H(X) \leq E(l_x) \leq H(X) + 1$.
  - Codage de longueur optimal !

Shannon et Huffman

- Codage associé à des arbres dyadiques (branches codées par 0 ou 1)
- Codage de Shannon-Fano (stratégie de haut en bas) :
  - ordonner les x par ordre de probabilité (initialisation),
  - couper récursivement en ensembles équiprobables (création d'un arbre)
  - coder x à l'aide des valeurs des branches de haut en bas : $l_x \leq \ulcorner \log_2 P_x \urcorner + 1$.
  - $H(X) \leq E(l_x) \leq H(X) + 2$
- Codage d'Huffman (stratégie de bas en haut) :
  - regrouper de manière récursive les deux symboles les moins probables (création d'une branche 0/1),
  - coder x à l'aide des valeurs des branches de haut en bas.
  - $H(X) \leq E(l_x) \leq H(X) + 1$.
  - Codage de longueur optimal !
- Limitation : perte de 1 bits par caractère et nécessité de refaire l'arbre si il y a un changement de proba.

Codage arithmétique

Codage arithmétique

- Passage par la fonction de répartition.

Codage arithmétique

- Passage par la fonction de répartition.
- Codage de Shannon-Fano-Elias :
  - On ordonne les x par ordre de probabilités décroissantes.
  - On code x à l'aide des $\ulcorner \log_2 P_x \urcorner$ premiers bits de $\sum_{x' \leq x} P_{x'} + P_x/2$
  - $H(X) \leq E(l_x) \leq H(X) + 1$
  - $\simeq$ Shannon-Fano, moins bon mais plus flexible...
  - Même algorithme pour $x \in \mathcal{X}^n$
    $H(X) \leq \frac{1}{n} E(l_x) \leq H(X) + 1/n$

Codage arithmétique

- Passage par la fonction de répartition.
- Codage de Shannon-Fano-Elias :
  - On ordonne les x par ordre de probabilités décroissantes.
  - On code x à l'aide des $\lceil \log_2 P_x \rceil$ premiers bits de $\sum_{x' \leq x} P_{x'} + P_x/2$
  - $H(X) \leq E(l_x) \leq H(X) + 1$
  - $\simeq$ Shannon-Fano, moins bon mais plus flexible...
  - Même algorithme pour $x \in \mathcal{X}^n$
    $H(X) \leq \frac{1}{n} E(l_x) \leq H(X) + 1/n$
- Codage arithmétique
  - Même algorithme avec l'ordre lexicographique : ordre indépendant des P_x.
  - On code $\mathbf{x} = x_1 \cdots x_n$ à l'aide des $\lceil \sum_{i=1}^{n} \log_2 P_x \rceil + 1$ premiers bits de $\sum_{x'_1 \cdots x'_n \leq x_1 \cdots x'_n} P_{x'} + P_x/2$
  - $H(X) + 1/n \leq \frac{1}{n} E(l_x) \leq H(X) + 2/n$
  - $\simeq$ Shannon-Fano-Elias, un peu moins bon mais beaucoup plus flexible...

Approche dictionnaire

Approche dictionnaire

- Problème les P_x sont inconnus le plus souvent...
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_x.

Approche dictionnaire

- Problème les P_X sont inconnus le plus souvent...
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_X.
- Approche de type dictionnaire :
  - créer un dictionnaire à partir de la liste de symbole à coder,
  - coder des groupes de symbole par leur position dans ce dictionnaire

Approche dictionnaire

- Problème les P_X sont inconnus le plus souvent. . .
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_X.
- Approche de type dictionnaire :
  - créer un dictionnaire à partir de la liste de symbole à coder,
  - coder des groupes de symbole par leur position dans ce dictionnaire
- Variations dans la création du dictionnaire.

Approche dictionnaire

- Problème les P_x sont inconnus le plus souvent...
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_x.
- Approche de type dictionnaire :
  - créer un dictionnaire à partir de la liste de symbole à coder,
  - coder des groupes de symbole par leur position dans ce dictionnaire
- Variations dans la création du dictionnaire.
- 1977- : LZW,LZ...,ZIP,ARJ,...

Approche dictionnaire

- Problème les P_x sont inconnus le plus souvent...
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_x.
- Approche de type dictionnaire :
  - créer un dictionnaire à partir de la liste de symbole à coder,
  - coder des groupes de symbole par leur position dans ce dictionnaire
- Variations dans la création du dictionnaire.
- 1977- : LZW,LZ...,ZIP,ARJ,...
- Optimalité asymptotique $\frac{1}{n}E(l_x) \to H(X)$

Approche dictionnaire

- Problème les P_x sont inconnus le plus souvent...
- Méthode pour obtenir un codage asymptotiquement optimal sans connaître les P_x.
- Approche de type dictionnaire :
  - créer un dictionnaire à partir de la liste de symbole à coder,
  - coder des groupes de symbole par leur position dans ce dictionnaire
- Variations dans la création du dictionnaire.
- 1977- : LZW,LZ...,ZIP,ARJ,...
- Optimalité asymptotique $\frac{1}{n}E(l_x) \to H(X)$
- Méthode utilisée dans le format GIF.

PNG

PNG

- Algorithme LZW breveté : besoin d’une solution de remplacement (95).

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62

12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

- Algorithme LZW breveté : besoin d’une solution de remplacement (95).
- Liste des pixels = liste de valeur d’intensité lumineuse.

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62

12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

$\downarrow$ prédiction + différence

- Algorithme LZW breveté : besoin d'une solution de remplacement (95).
- Liste des pixels = liste de valeur d'intensité lumineuse.
- Prédiction possible d'une valeur en fonction de celles déjà vues et codage de l'erreur de prédiction.

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62
12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

$\downarrow$ prédiction + différence

12	-1	4	2	3	-2	-1	-1	-6	1	3	13	-6	-4	-1	0	-4	0	5	2	-1	1	-3	3	-6	0	2	4	-1	1	17	26
-50	2	5	1	0	3	9	31	-39	4	2	2	2	13	19	6	-45	3	2	1	2	48	3	-2	-44	4	1	2	1	32	6	1

- Algorithme LZW breveté : besoin d'une solution de remplacement (95).
- Liste des pixels = liste de valeur d'intensité lumineuse.
- Prédiction possible d'une valeur en fonction de celles déjà vues et codage de l'erreur de prédiction.

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62
12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

$\downarrow$ prédiction + différence

12	-1	4	2	3	-2	-1	-1	-6	1	3	13	-6	-4	-1	0	-4	0	5	2	-1	1	-3	3	-6	0	2	4	-1	1	17	26
-50	2	5	1	0	3	9	31	-39	4	2	2	2	13	19	6	-45	3	2	1	2	48	3	-2	-44	4	1	2	1	32	6	1

- Algorithme LZW breveté : besoin d'une solution de remplacement (95).
- Liste des pixels = liste de valeur d'intensité lumineuse.
- Prédiction possible d'une valeur en fonction de celles déjà vues et codage de l'erreur de prédiction.
- Modèle simple : valeur précédente, moyenne locale,. . .

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62
12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

$\downarrow$ prédiction + différence

12	-1	4	2	3	-2	-1	-1	-6	1	3	13	-6	-4	-1	0	-4	0	5	2	-1	1	-3	3	-6	0	2	4	-1	1	17	26
-50	2	5	1	0	3	9	31	-39	4	2	2	2	13	19	6	-45	3	2	1	2	48	3	-2	-44	4	1	2	1	32	6	1

- Algorithme LZW breveté : besoin d'une solution de remplacement (95).
- Liste des pixels = liste de valeur d'intensité lumineuse.
- Prédiction possible d'une valeur en fonction de celles déjà vues et codage de l'erreur de prédiction.
- Modèle simple : valeur précédente, moyenne locale,. . .
- Utilisation d'une meilleure modélisation statistique d'une image.

PNG

12	13	15	17	20	18	17	16	10	11	14	27	21	17	16	16	12	12	17	19	18	19	16	19	13	13	15	19	18	19	36	62
12	14	19	20	20	23	32	63	14	18	20	22	24	37	56	62	17	20	22	23	25	63	66	64	20	24	25	27	28	60	66	67

$\downarrow$ prédiction + différence

12	-1	4	2	3	-2	-1	-1	-6	1	3	13	-6	-4	-1	0	-4	0	5	2	-1	1	-3	3	-6	0	2	4	-1	1	17	26
-50	2	5	1	0	3	9	31	-39	4	2	2	2	13	19	6	-45	3	2	1	2	48	3	-2	-44	4	1	2	1	32	6	1

- Algorithme LZW breveté : besoin d'une solution de remplacement (95).
- Liste des pixels = liste de valeur d'intensité lumineuse.
- Prédiction possible d'une valeur en fonction de celles déjà vues et codage de l'erreur de prédiction.
- Modèle simple : valeur précédente, moyenne locale,. . .
- Utilisation d'une meilleure modélisation statistique d'une image.
- $\Longrightarrow$ Amélioration de l'efficacité des algorithmes de type dictionnaire.

Modélisation statistique

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.
- Importance de construire des modèles statistiques adaptés.

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.
- Importance de construire des modèles statistiques adaptés.
- Modèle de distributions simples (iid avec une distribution connue) : *OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL*

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.
- Importance de construire des modèles statistiques adaptés.
- Modèle de distributions simples (iid avec une distribution connue) :
  OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
- Modèle de dépendance en fonction du passé :
  IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.
- Importance de construire des modèles statistiques adaptés.
- Modèle de distributions simples (iid avec une distribution connue) :
  OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
- Modèle de dépendance en fonction du passé :
  IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
- Modèle de distribution plus complexes :
  THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED

Modélisation statistique

- Le facteur de compression dépend de la probabilité dans le modèle.
- Importance de construire des modèles statistiques adaptés.
- Modèle de distributions simples (iid avec une distribution connue) :
  OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
- Modèle de dépendance en fonction du passé :
  IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
- Modèle de distribution plus complexes :
  THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED
- Pour les images $\simeq$ facteur 4 de compression avec les meilleurs modèles (Markovien avec apprentissage. . .).

Entropie des processus stationnaire

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)
- Théorie des séquences iids s'appliquent asymptotiquement...

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)
- Théorie des séquences iids s'appliquent asymptotiquement...
- Ex : modélisation markovienne via $P(X_n | X_1 = x_1, \ldots, X_{n-1} = x_{n-1})$

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)
- Théorie des séquences iids s'appliquent asymptotiquement...
- Ex : modélisation markovienne via $P(X_n | X_1 = x_1, \ldots, X_{n-1} = x_{n-1})$
- Utilisation de l'algorithme de compression arithmétique possible sans modification.

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)
- Théorie des séquences iids s'appliquent asymptotiquement...
- Ex : modélisation markovienne via $P(X_n | X_1 = x_1, \ldots, X_{n-1} = x_{n-1})$
- Utilisation de l'algorithme de compression arithmétique possible sans modification.
- Approche dictionnaire encore valide dans ce cadre...

Entropie des processus stationnaire

- Processus stationnaire : $(X_1, \ldots, X_k) \sim (X_{1+l}, \ldots, X_{k+l})$ pour toute longueur k et décalage l.
- Entropie : $H(X) = \lim_n \frac{1}{n} H(X_1, \ldots, X_n)$ (si elle existe...)
- Prop : $H(X) = \lim_n H(X_n | X_1, \ldots, X_{n-1})$ (si elle existe...)
- Théorie des séquences iids s'appliquent asymptotiquement...
- Ex : modélisation markovienne via $P(X_n | X_1 = x_1, \ldots, X_{n-1} = x_{n-1})$
- Utilisation de l'algorithme de compression arithmétique possible sans modification.
- Approche dictionnaire encore valide dans ce cadre...
- En pratique : dictionnaire lorsqu'on a pas de modèles explicites sur les probabilités (textes, fichiers...) et arithmétique lorsqu'il y a un modèle (images, sons...).

Compression avec perte

Compression avec perte

 $\rightarrow$  $\rightleftarrows$ 0110101...01...

- Taux de compression sans perte souvent insuffisant.

Compression avec perte

 $\rightarrow$  $\rightleftarrows$ 0110101...

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !

Compression avec perte

 $\rightarrow$   $\rightleftarrows$ 011010

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !
- Exemples (facteur 16 de compression) :

Compression avec perte

$\rightarrow \qquad \rightleftarrows$ 0110101...

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !
- Exemples (facteur 16 de compression) :
  - changement de résolution,

Compression avec perte

 $\rightarrow$  $\rightleftarrows$ 0110101...

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !
- Exemples (facteur 16 de compression) :
  - changement de résolution,
  - nombre de couleurs utilisées,

Compression avec perte

$\rightarrow$

$\rightleftarrows$ 0110101...

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !
- Exemples (facteur 16 de compression) :
  - changement de résolution,
  - nombre de couleurs utilisées,
  - facteur de qualité (JPEG)

Compression avec perte

 $\rightarrow$  $\rightleftarrows$ 0110101...

- Taux de compression sans perte souvent insuffisant.
- Pour améliorer le taux de compression, il faut perdre de l'information !
- Exemples (facteur 16 de compression) :
  - changement de résolution,
  - nombre de couleurs utilisées,
  - facteur de qualité (JPEG)
- Comment ça marche JPEG ?

JPEG

JPEG

- Algorithme proposé en 1990 par un comité d'expert (Joint Photographic Experts Group).

JPEG

- Algorithme proposé en 1990 par un comité d'expert (Joint Photographic Experts Group).
- Principes :
  - Quantification après un changement de base (DCT).
  - Codage statistique (Huffman).

JPEG

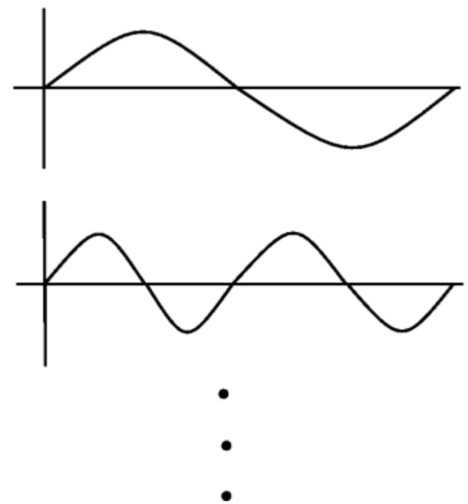
- Algorithme proposé en 1990 par un comité d'expert (Joint Photographic Experts Group).
- Principes :
  - Quantification après un changement de base (DCT).
  - Codage statistique (Huffman).
- Certain succès !

Base de Fourier

Base de Fourier

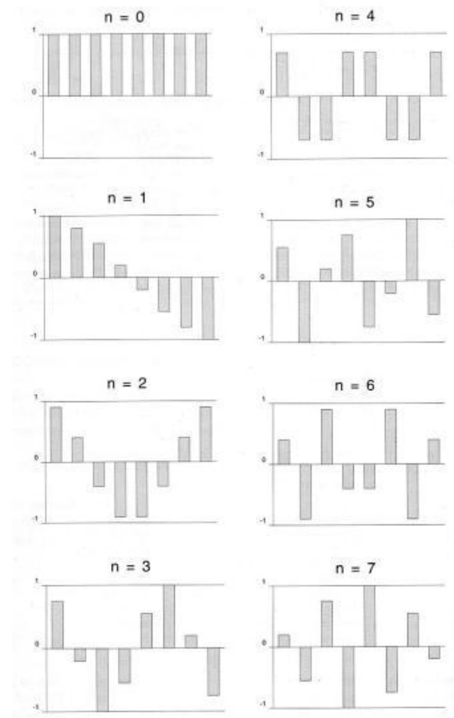
- Joseph Fourier : La propagation de la chaleur dans les solides (1807).

Base de Fourier



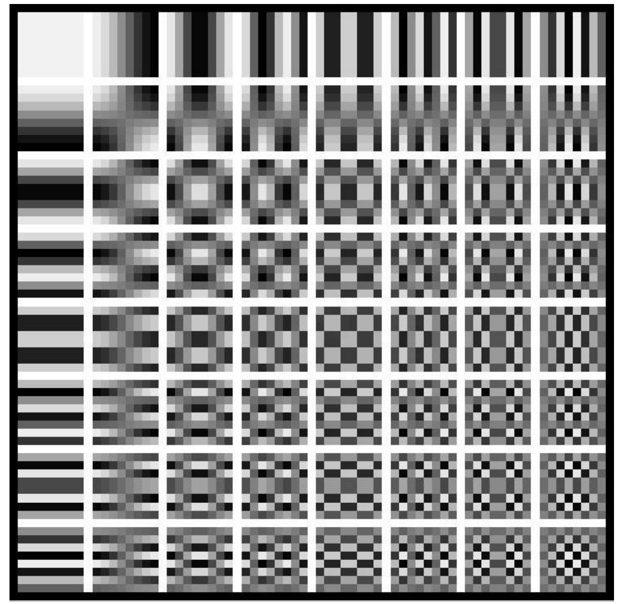
- Joseph Fourier : La propagation de la chaleur dans les solides (1807).
- Décomposition des fonctions sur l'intervalle dans une base de cosinus et de sinus.

Base de Fourier



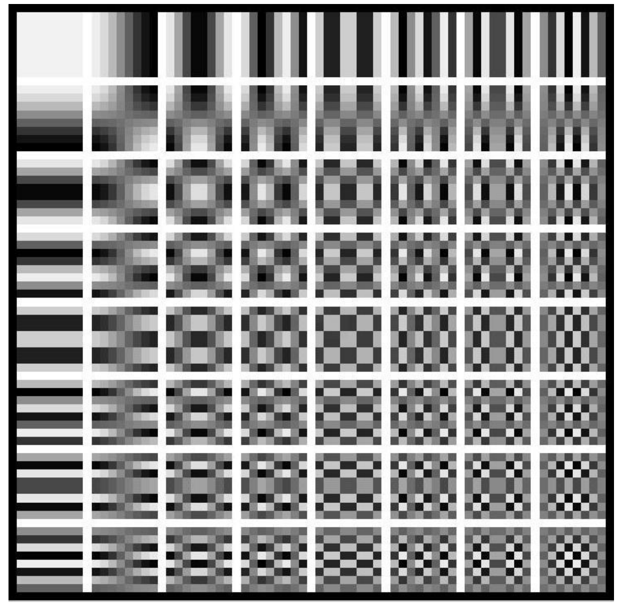
- Joseph Fourier : La propagation de la chaleur dans les solides (1807).
- Décomposition des fonctions sur l'intervalle dans une base de cosinus et de sinus.
- Équivalent discret pour les vecteurs à N éléments.

Base de Fourier



- Joseph Fourier : La propagation de la chaleur dans les solides (1807).
- Décomposition des fonctions sur l'intervalle dans une base de cosinus et de sinus.
- Équivalent discret pour les vecteurs à N éléments.
- DCT 2D.

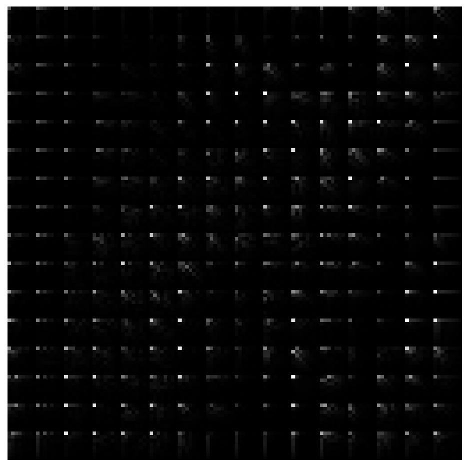
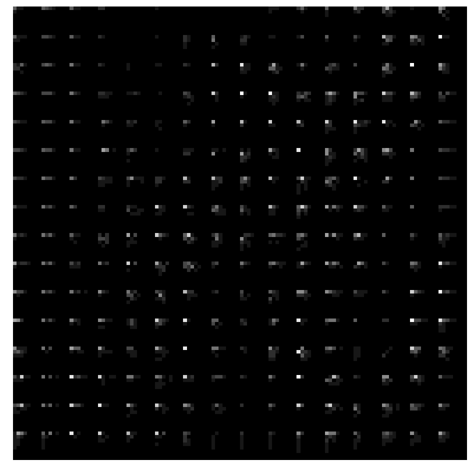
Base de Fourier



- Joseph Fourier : La propagation de la chaleur dans les solides (1807).
- Décomposition des fonctions sur l'intervalle dans une base de cosinus et de sinus.
- Équivalent discret pour les vecteurs à N éléments.
- DCT 2D.
- Rien qu'un changement de base vers une base plus adaptée. (Algèbre linéaire)

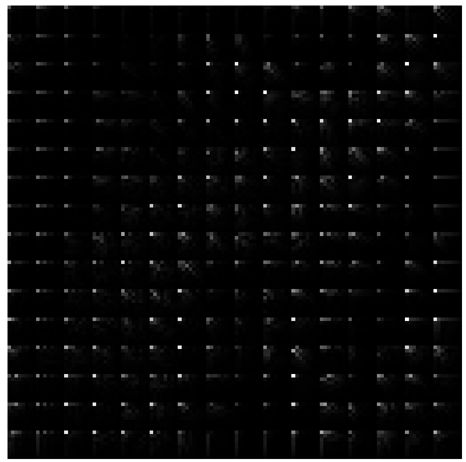
Quantification

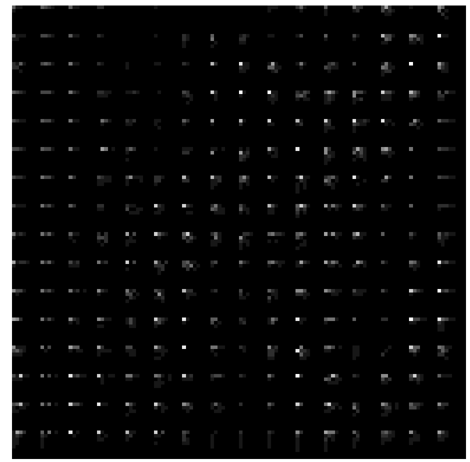
Quantification

- Réduction du nombre de symboles utilisés.

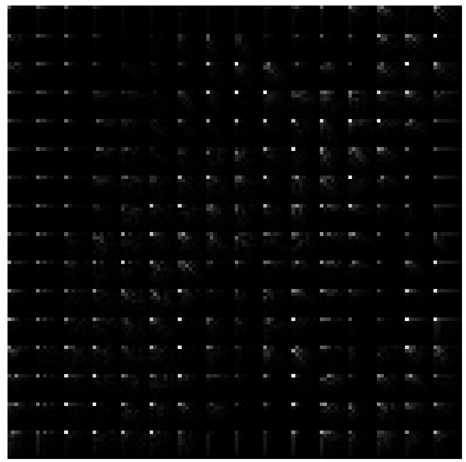
Quantification

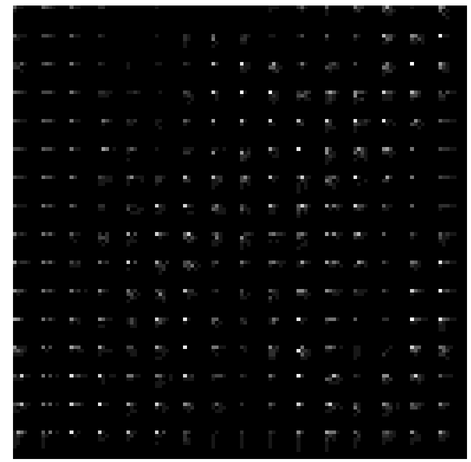




- Réduction du nombre de symboles utilisés.
- $Q_\Delta(x) = \lfloor x/\Delta + .5 \rfloor \Delta$ (arrondi).

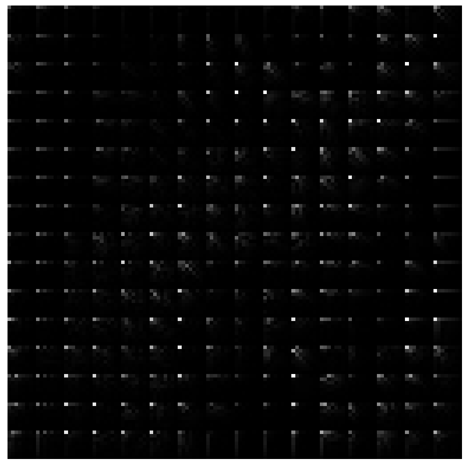
Quantification

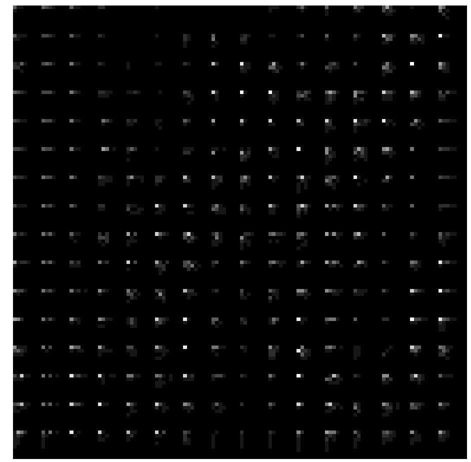




- Réduction du nombre de symboles utilisés.
- $Q_\Delta(x) = \lfloor x/\Delta + .5 \rfloor \Delta$ (arrondi).
- On code uniquement $\lfloor x/\Delta + .5 \rfloor$ pour Δ connu (facteur de qualité).

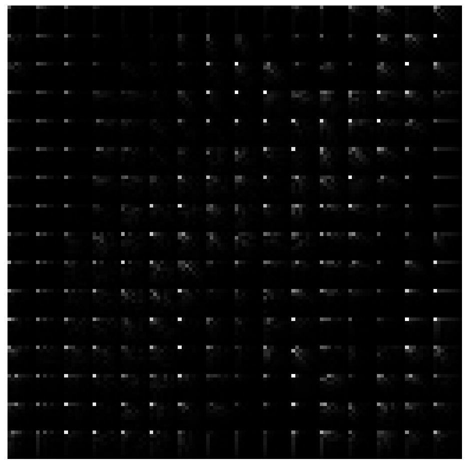
Quantification

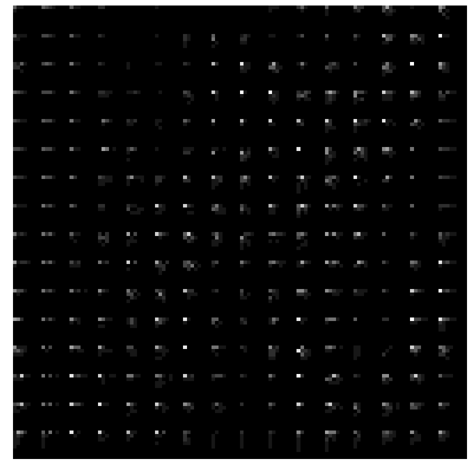




- Réduction du nombre de symboles utilisés.
- $Q_\Delta(x) = \lfloor x/\Delta + .5 \rfloor \Delta$ (arrondi).
- On code uniquement $\lfloor x/\Delta + .5 \rfloor$ pour Δ connu (facteur de qualité).
- Erreur quadratique par coefficient bornée par $\Delta^2/4$.

Quantification





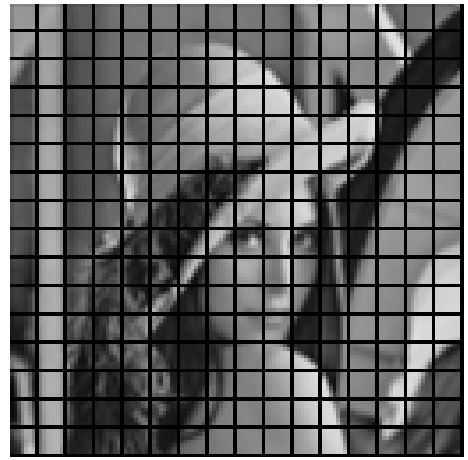
- Réduction du nombre de symboles utilisés.
- $Q_\Delta(x) = \lfloor x/\Delta + .5 \rfloor \Delta$ (arrondi).
- On code uniquement $\lfloor x/\Delta + .5 \rfloor$ pour Δ connu (facteur de qualité).
- Erreur quadratique par coefficient bornée par $\Delta^2/4$.
- Pour les coefficients très petits, erreurs très petites également !

JPEG

JPEG

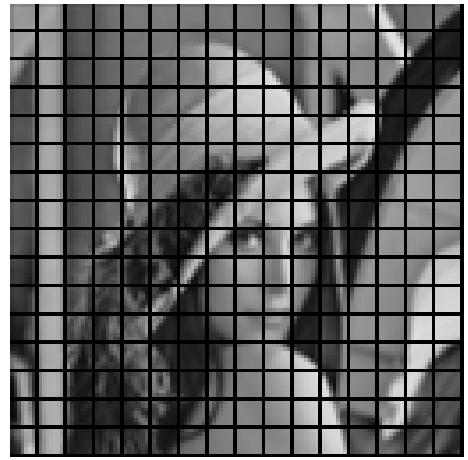
- Image initiale.

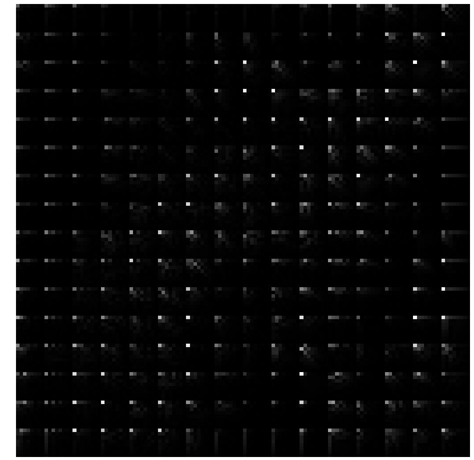
JPEG



- Image initiale.
- Découpage en carré 8×8.

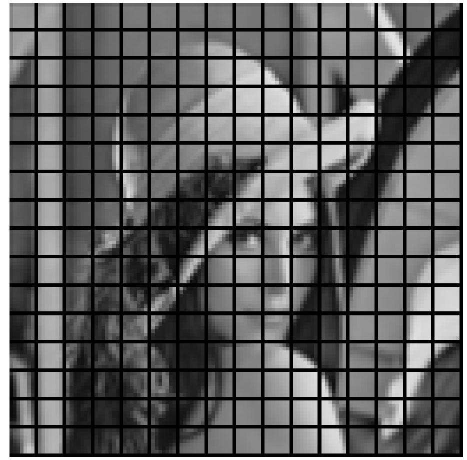
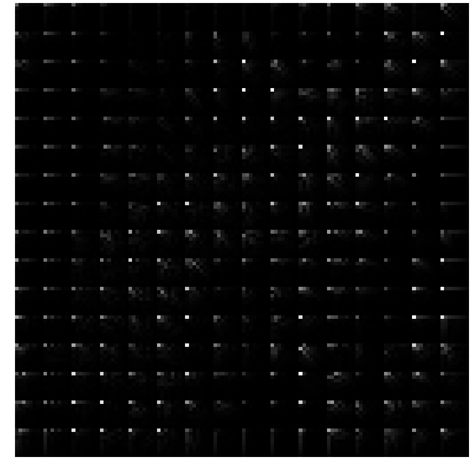
JPEG





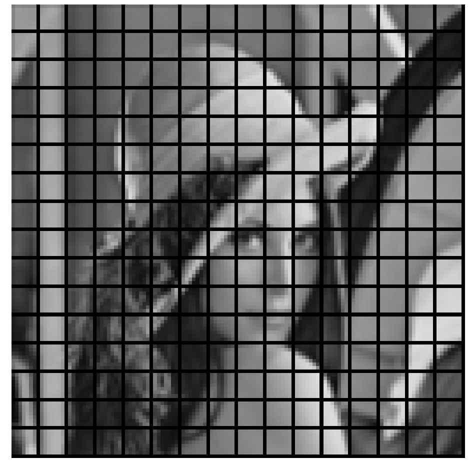
- Image initiale.
- Découpage en carré 8×8.
- Transformation linéaire (changement de base).

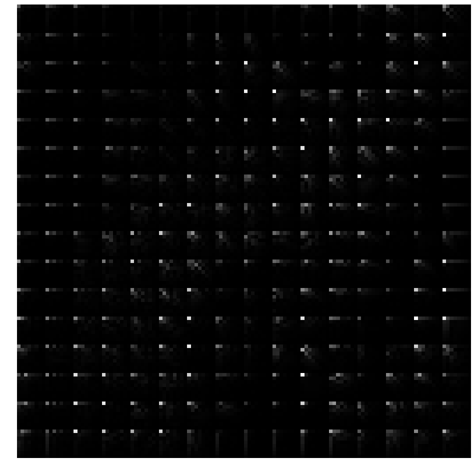
JPEG

- Image initiale.
- Découpage en carré 8×8.
- Transformation linéaire (changement de base).
- Beaucoup de petits coefficients.

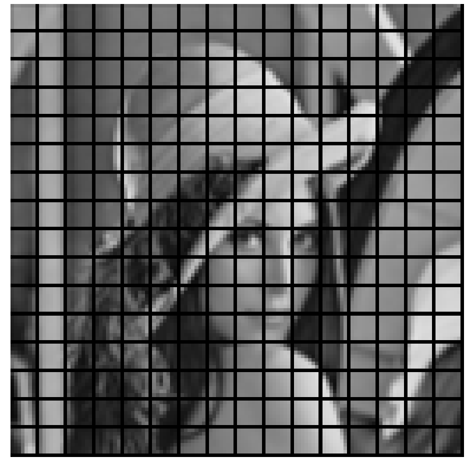
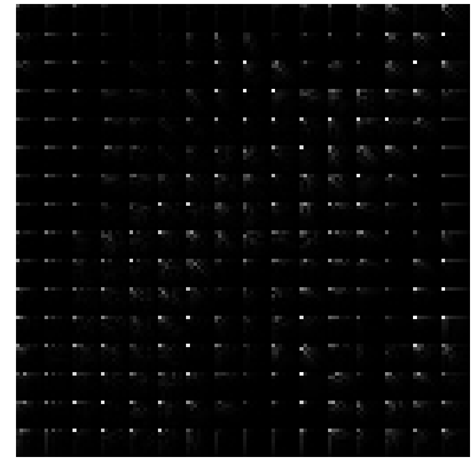
JPEG





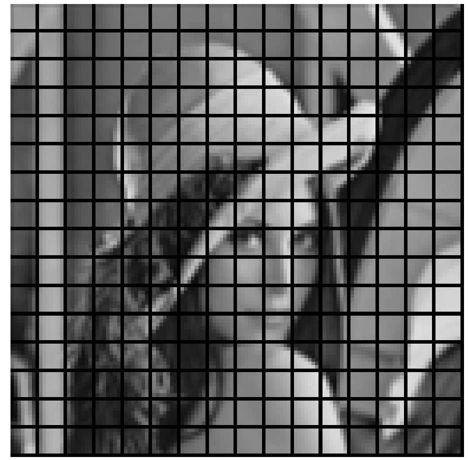
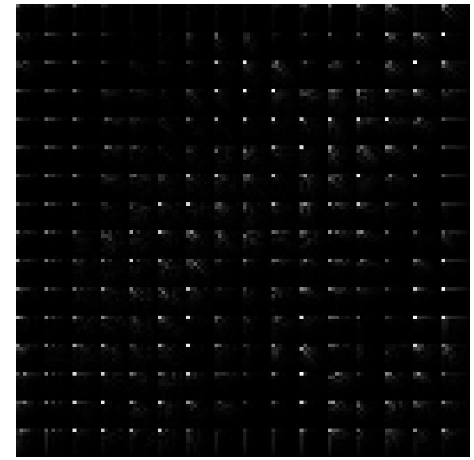
- Image initiale.
- Découpage en carré 8×8.
- Transformation linéaire (changement de base).
- Beaucoup de petits coefficients.
- Quantification de ces coefficients.

JPEG

- Image initiale.
- Découpage en carré 8×8.
- Transformation linéaire (changement de base).
- Beaucoup de petits coefficients.
- Quantification de ces coefficients.
- Compression sans perte par un codage de Huffman de ces coefficients quantifiés.

JPEG

- Image initiale.
- Découpage en carré 8×8.
- Transformation linéaire (changement de base).
- Beaucoup de petits coefficients.
- Quantification de ces coefficients.
- Compression sans perte par un codage de Huffman de ces coefficients quantifiés.
- Si beaucoup de coefficients sont quantifiés à 0, bon taux de compression...

Approximation dans une base orthonormée

Approximation dans une base orthonormée

- Décomposition dans une base orthonormée $\mathbf{B} = \{g_m\}_{m \in \mathbb{N}}$

$$f = \sum_{m \in \mathbb{N}} \langle f, g_m \rangle \, g_m \ .$$

Approximation dans une base orthonormée

- Décomposition dans une base orthonormée $\mathbf{B} = \{g_m\}_{m \in \mathbb{N}}$

$$f = \sum_{m \in \mathbb{N}} \langle f, g_m \rangle \, g_m \, .$$

- Approximation avec M vecteurs choisis adaptativement

$$f_M = \sum_{m \in I_M} \langle f, g_m \rangle \, g_m \, .$$

Approximation dans une base orthonormée

- Décomposition dans une base orthonormée $\mathbf{B} = \{g_m\}_{m \in \mathbb{N}}$

$$f = \sum_{m \in \mathbb{N}} \langle f, g_m \rangle \, g_m \ .$$

- Approximation avec M vecteurs choisis adaptativement

$$f_M = \sum_{m \in I_M} \langle f, g_m \rangle \, g_m \, .$$

- Pour minimiser $\quad \|f - f_M\|^2 = \sum_{m \notin I_M} |\langle f, g_m \rangle|^2,$

  sélection des M plus grands produits scalaires :

$$I_M = \{m, \ |\langle f, g_m \rangle| > T_M\} \text{ : seuillage.}$$

Approximation dans une base orthonormée

- Décomposition dans une base orthonormée $\mathbf{B} = \{g_m\}_{m \in \mathbb{N}}$

$$f = \sum_{m \in \mathbb{N}} \langle f, g_m \rangle \, g_m \; .$$

- Approximation avec M vecteurs choisis adaptativement

$$f_M = \sum_{m \in I_M} \langle f, g_m \rangle \, g_m \, .$$

- Pour minimiser $\quad \|f - f_M\|^2 = \sum_{m \notin I_M} |\langle f, g_m \rangle|^2,$

  sélection des M plus grands produits scalaires :

$$I_M = \{m, \; |\langle f, g_m \rangle| > T_M\} \text{ : seuillage.}$$

- Situation similaire pour la compression.

Compression par transformée

Compression par transformée

- Compression par transformée = transformation + quantification + codage entropique.

Compression par transformée

- Compression par transformée = transformation + quantification + codage entropique.
- $$f_R = \sum Q(\langle f, g_m \rangle) g_m = \sum_{Q(\langle f, g_m \rangle) \neq 0} Q(\langle f, g_m \rangle)\, g_m \quad .$$

Compression par transformée

- Compression par transformée = transformation + quantification + codage entropique.
- $$f_R = \sum Q(\langle f, g_m \rangle) g_m = \sum_{Q(\langle f, g_m \rangle) \neq 0} Q(\langle f, g_m \rangle)\, g_m \quad .$$
- $$\|f - f_R\|^2 = \sum_{Q(\langle f, g_m \rangle) \neq 0} (\langle f, g_m \rangle - Q(\langle f, g_m \rangle))^2 + \sum_{Q(\langle f, g_m \rangle) = 0} \langle f, g_m \rangle^2 \quad .$$

Compression par transformée

- Compression par transformée = transformation + quantification + codage entropique.

- $$f_R = \sum Q(\langle f, g_m \rangle) g_m = \sum_{Q(\langle f, g_m \rangle) \neq 0} Q(\langle f, g_m \rangle)\, g_m \quad .$$

- $$\begin{aligned} \|f - f_R\|^2 = & \sum_{Q(\langle f, g_m \rangle) \neq 0} (\langle f, g_m \rangle - Q(\langle f, g_m \rangle))^2 \\ & + \sum_{Q(\langle f, g_m \rangle) = 0} \langle f, g_m \rangle^2 \quad . \end{aligned}$$

- Erreur de quantification + Erreur d'approximation.

Compression par transformée

- Compression par transformée = transformation + quantification + codage entropique.
- $$f_R = \sum Q(\langle f, g_m\rangle) g_m = \sum_{Q(\langle f, g_m\rangle)\neq 0} Q(\langle f, g_m\rangle)\, g_m \quad .$$
- $$\|f - f_R\|^2 = \sum_{Q(\langle f, g_m\rangle)\neq 0} (\langle f, g_m\rangle - Q(\langle f, g_m\rangle))^2 + \sum_{Q(\langle f, g_m\rangle)= 0} \langle f, g_m\rangle^2 \quad .$$
- Erreur de quantification + Erreur d'approximation.
- Quantification uniforme de pas Δ avec boite 0 de taille double :
  - $\|f - f_R\|^2 = M\Delta^2/12 + \|f - f_M\||^2$
  - $R \simeq M(C_1 + C_2 \log(M/N))$.

Choix de la base

Choix de la base

- Ne reste que le choix de la base…

Choix de la base

- Ne reste que le choix de la base...
- Théorie de l'approximation.

Choix de la base

- Ne reste que le choix de la base...
- Théorie de l'approximation.
- Th : $\forall \alpha > 0$, $\forall f$,

$$\exists C, \quad \forall M, \quad \|f - f_M\|^2 \leq C M^{-\alpha}$$

$$\Leftrightarrow \exists C', \quad \forall T > 0, \quad \inf_I \|f - f_I\|^2 + T^2 |I| \leq C' (T^2)^{\alpha/(\alpha+1)}$$

Choix de la base

- Ne reste que le choix de la base...
- Théorie de l'approximation.
- Th : $\forall \alpha > 0, \forall f,$

$$\exists C, \quad \forall M, \quad \|f - f_M\|^2 \leq C M^{-\alpha}$$

$$\Leftrightarrow \exists C', \quad \forall T > 0, \quad \inf_I \|f - f_I\|^2 + T^2|I| \leq C'(T^2)^{\alpha/(\alpha+1)}$$

- Preuve :

$\Rightarrow$:

$$\begin{aligned} \inf_I \|f - f_I\|^2 + T^2|I| &\leq \inf_M \|f - f_M\|^2 + T^2 M \leq \inf_M C M^{-\alpha} + T^2 M \\ &\leq \inf_{x>0} C x^{-\alpha} + T^2(x+1) \\ &\leq C\left(T^2/(2\alpha C)\right)^{\alpha/(\alpha+1)} + T^2\left(T^2/(2\alpha C)\right)^{-1/(\alpha+1)} \\ &\leq C'(T^2)^{\alpha/(\alpha+1)} \end{aligned}$$

$\Leftarrow$: $\forall T, \quad \exists I_T$, tel que $\quad \|f - f_{I_T}\|^2 + T^2|I_T| \leq C'(T^2)^{\alpha/(\alpha+1)}$

donc pour $M = |I_T|$ $\quad \|f - f_M\|^2 \leq \|f - f_{I_T}\|^2 \leq C'(T^2)^{\alpha/(\alpha+1)}$ et $T^2 M \leq C'(T^2)^{\alpha/(\alpha+1)}$

$$\Rightarrow \|f - f_M\|^2 \leq C'' M^{-\alpha}$$

Choix de la base

- Ne reste que le choix de la base...
- Théorie de l'approximation.
- Th : $\forall \alpha > 0$, $\forall f$,

$$\exists C, \quad \forall M, \quad \|f - f_M\|^2 \leq C M^{-\alpha}$$

$$\Leftrightarrow \exists C', \quad \forall T > 0, \quad \inf_I \|f - f_I\|^2 + T^2 |I| \leq C' (T^2)^{\alpha/(\alpha+1)}$$

- Preuve :

$\Rightarrow$:

$$\begin{aligned}\inf_I \|f - f_I\|^2 + T^2|I| &\leq \inf_M \|f - f_M\|^2 + T^2 M \leq \inf_M C M^{-\alpha} + T^2 M \\ &\leq \inf_{x>0} C x^{-\alpha} + T^2(x+1) \\ &\leq C \left(T^2/(2\alpha C)\right)^{\alpha/(\alpha+1)} + T^2 \left(T^2/(2\alpha C)\right)^{-1/(\alpha+1)} \\ &\leq C'(T^2)^{\alpha/(\alpha+1)}\end{aligned}$$

$\Leftarrow$: $\forall T$, $\quad \exists I_T$, tel que $\quad \|f - f_{I_T}\|^2 + T^2 |I_T| \leq C'(T^2)^{\alpha/(\alpha+1)}$

donc pour $M = |I_T|$

$$\|f - f_M\|^2 \leq \|f - f_{I_T}\|^2 \leq C'(T^2)^{\alpha/(\alpha+1)} \quad \text{et} \quad T^2 M \leq C'(T^2)^{\alpha/(\alpha+1)}$$

$$\Rightarrow \|f - f_M\|^2 \leq C'' M^{-\alpha}$$

- Amélioration possible si la base permet d'approcher plus efficacement les objets d'intérêts !

JPEG 2000

JPEG 2000

- *Nouveau standard* (2000. . .).

JPEG 2000

- *Nouveau standard* (2000. . .).
- Comité JPEG 2000 : premier jet (Décembre 2000), version définitive (Août 2002).

JPEG 2000

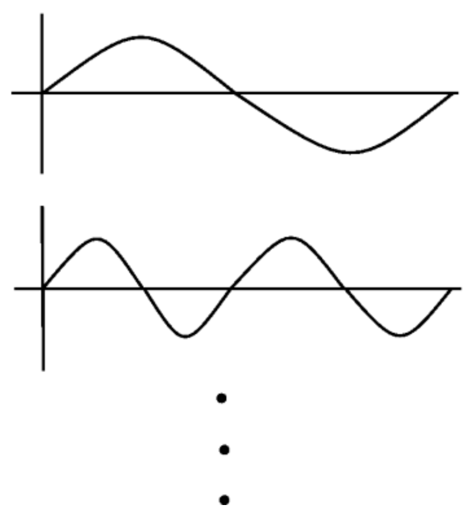
- *Nouveau standard* (2000. . .).
- Comité JPEG 2000 : premier jet (Décembre 2000), version définitive (Août 2002).
- Différences avec JPEG :
  - Autre base (ondelette),
  - Modélisation plus fine

JPEG 2000

- *Nouveau standard* (2000. . .).
- Comité JPEG 2000 : premier jet (Décembre 2000), version définitive (Août 2002).
- Différences avec JPEG :
  - Autre base (ondelette),
  - Modélisation plus fine
- Avantage :
  - Performance,
  - Usage (échelonnabilité,progressivité,. . .)

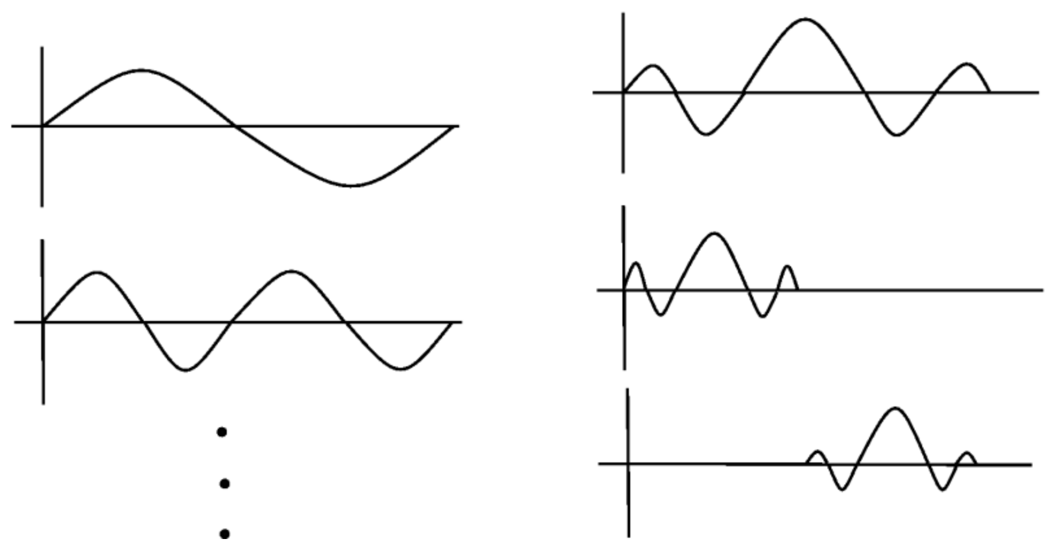
Ondelettes 1D

Ondelettes 1D



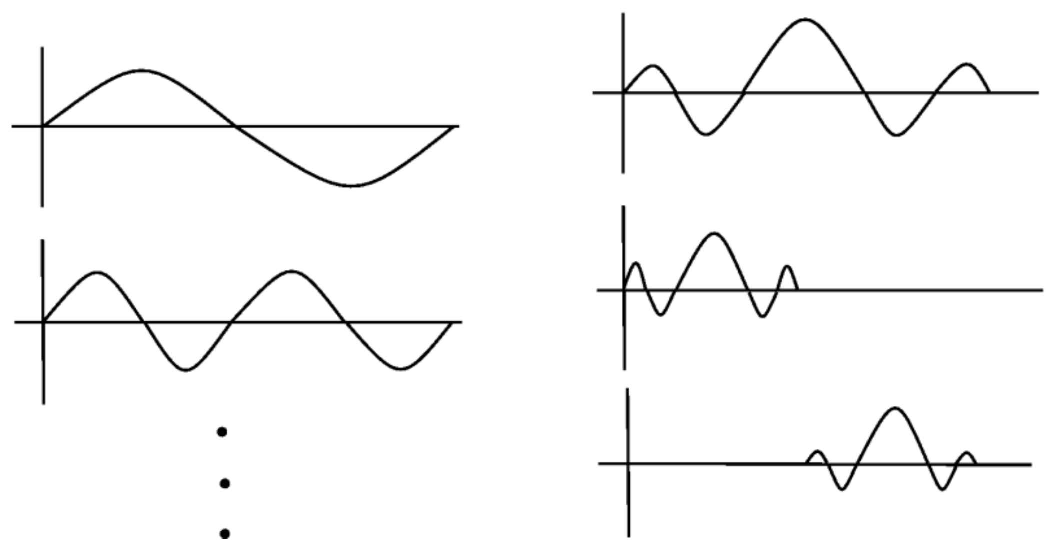
- Fourier : modèle stationnaire.

Ondelettes 1D



- Fourier : modèle stationnaire.
- Ondelettes : localisation.

Ondelettes 1D

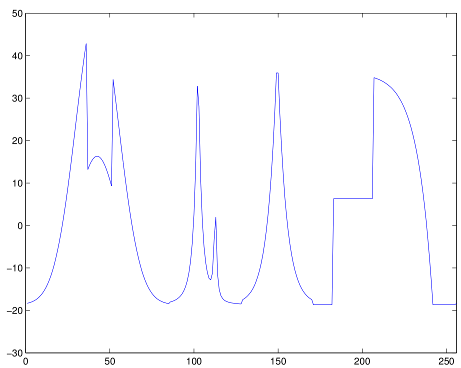


- Fourier : modèle stationnaire.
- Ondelettes : localisation.
- Structure multirésolution : approximations successives et détails.

Multirésolution

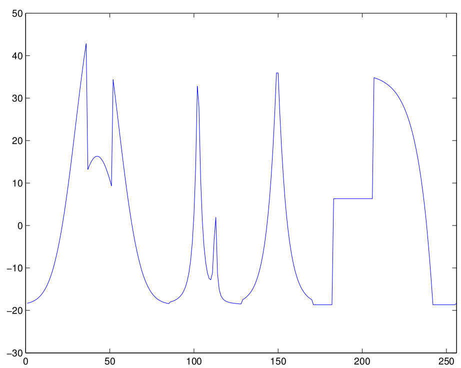
Multirésolution

256

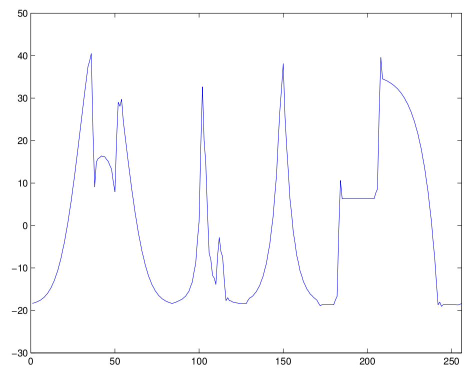


Multirésolution

256

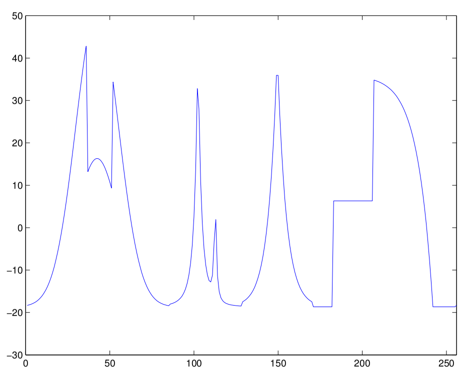


128

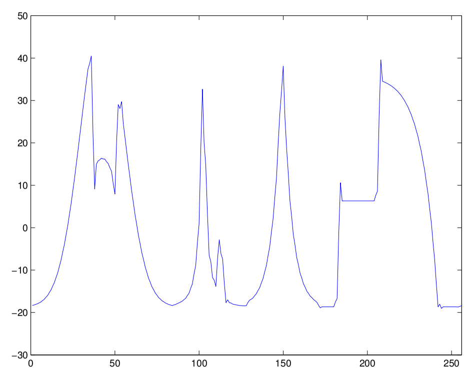


Multirésolution

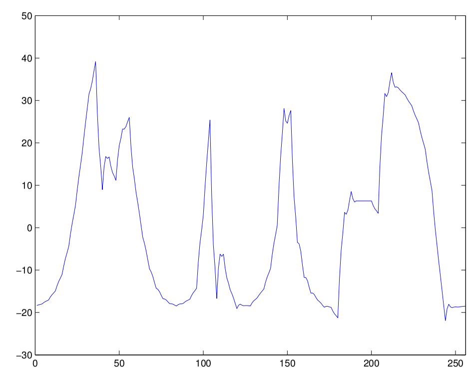
256



128

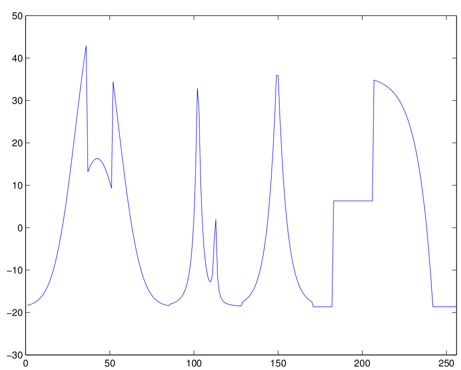


64

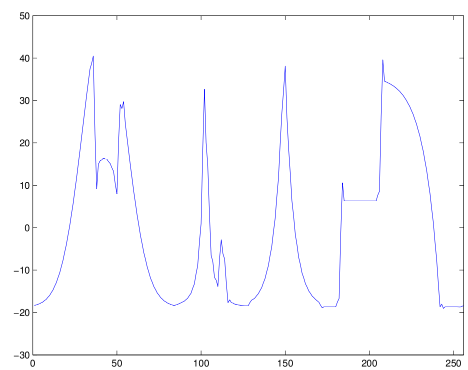


Multirésolution

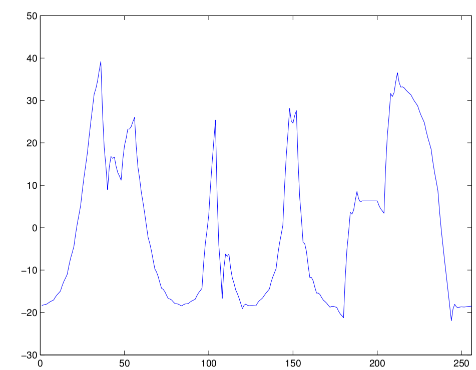
256



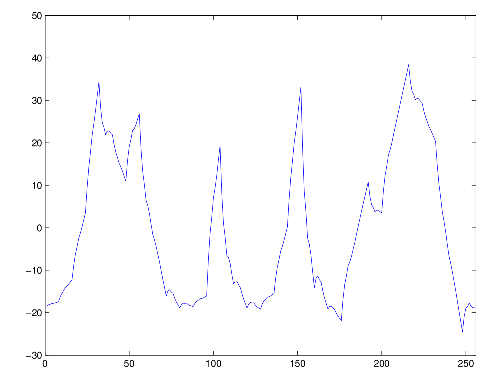
128



64

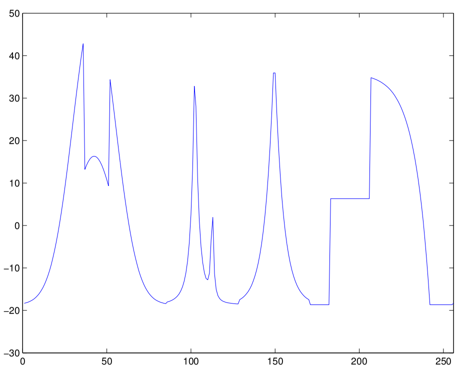


32

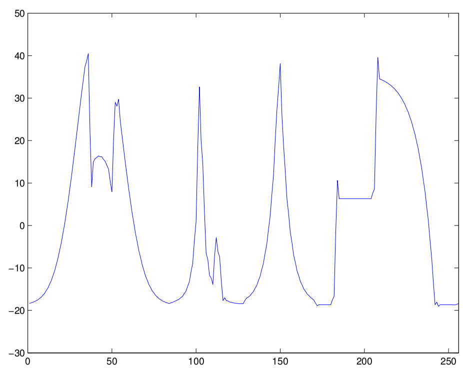


Multirésolution

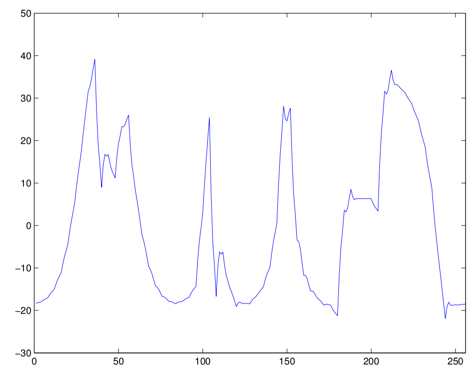
256



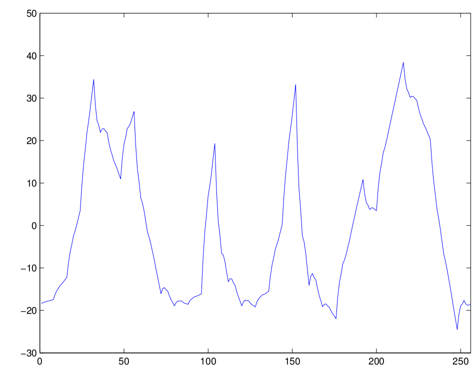
128

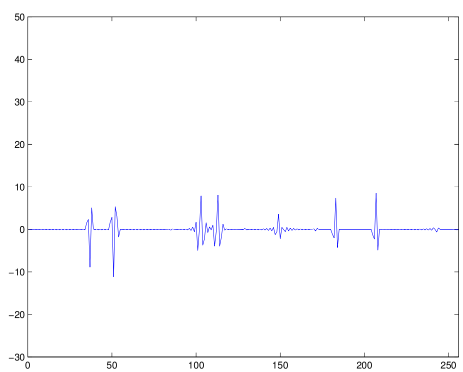


64

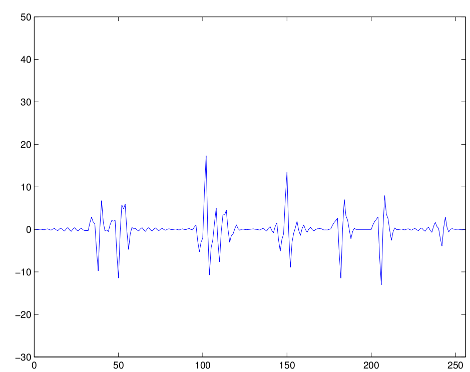


32

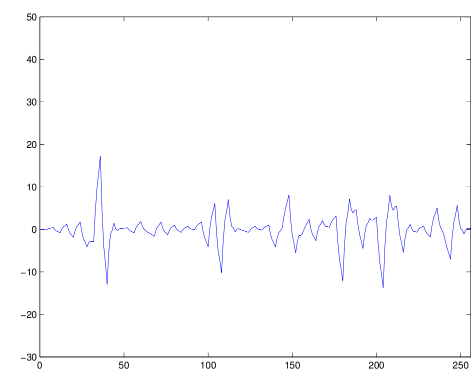




128



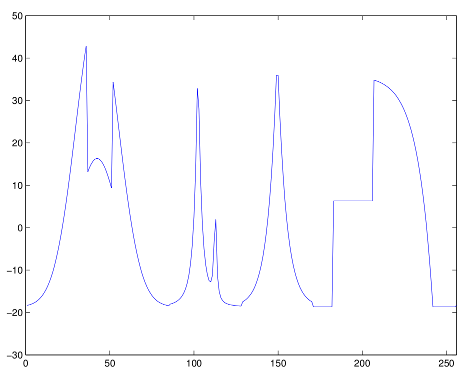
64



32

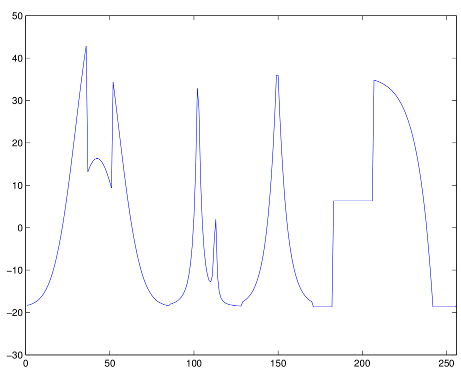
Multirésolution

256

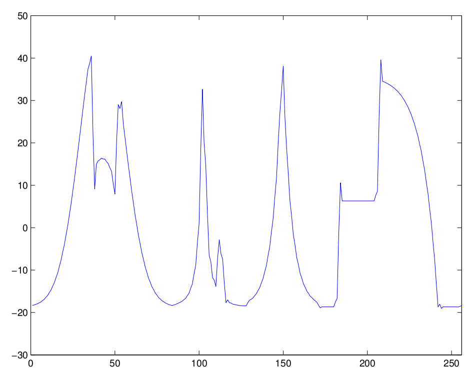


Multirésolution

256

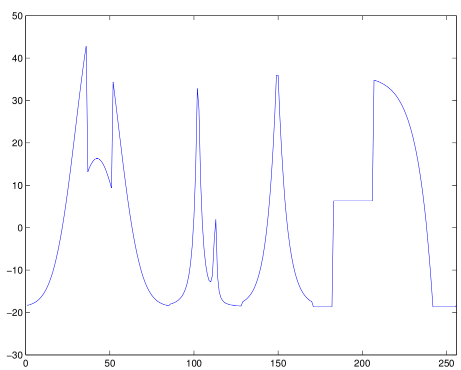


128

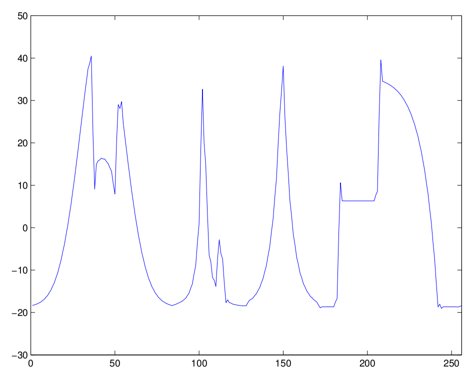


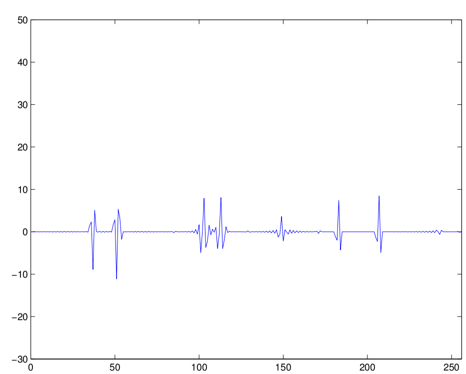
Multirésolution

256



128

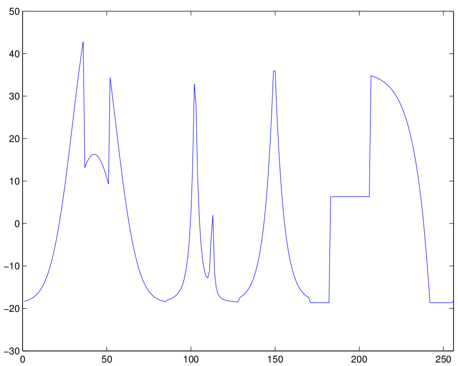




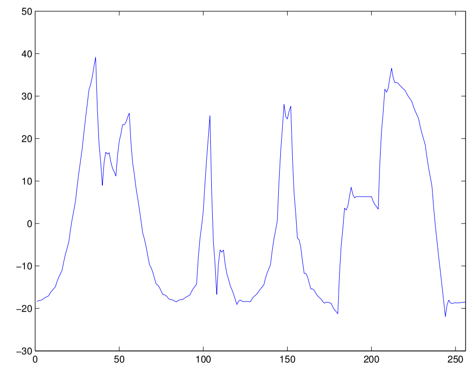
128

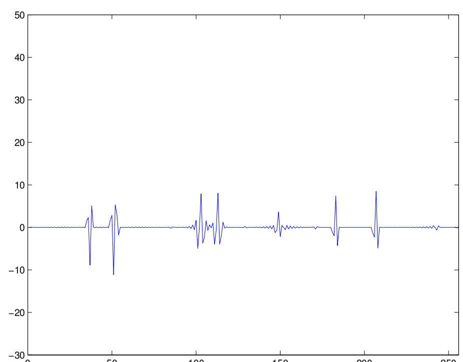
Multirésolution

256

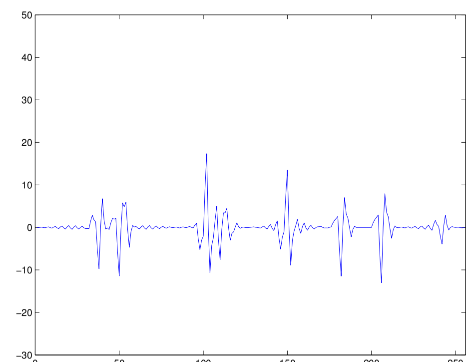


64





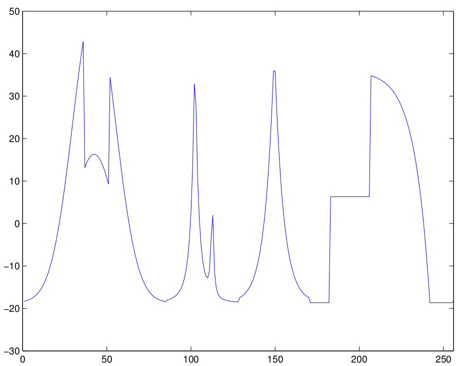
128



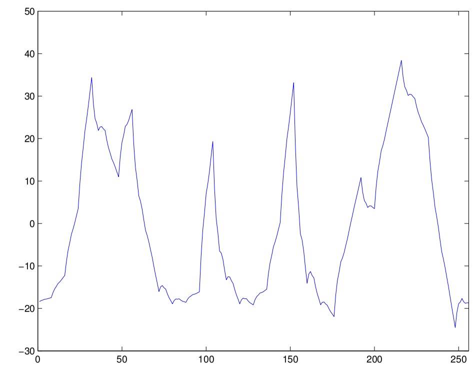
64

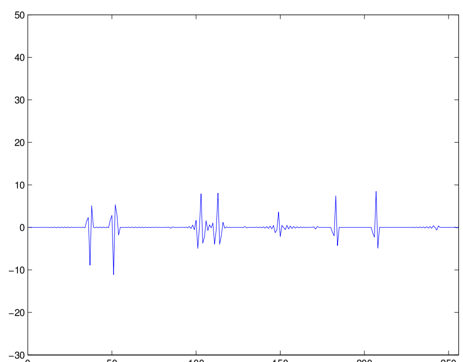
Multirésolution

256

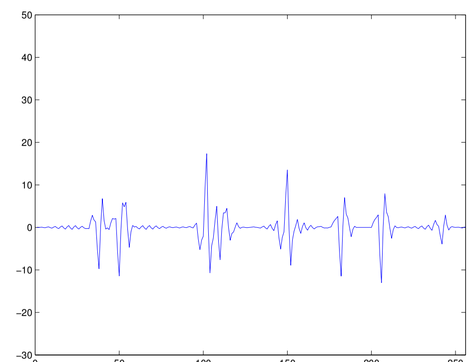


32

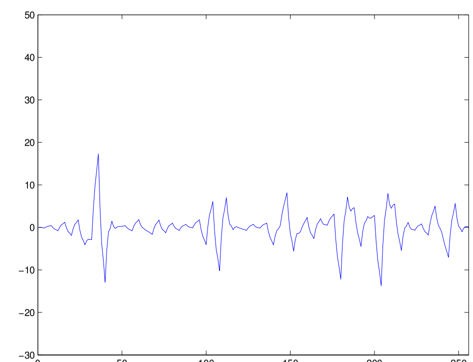




128



64

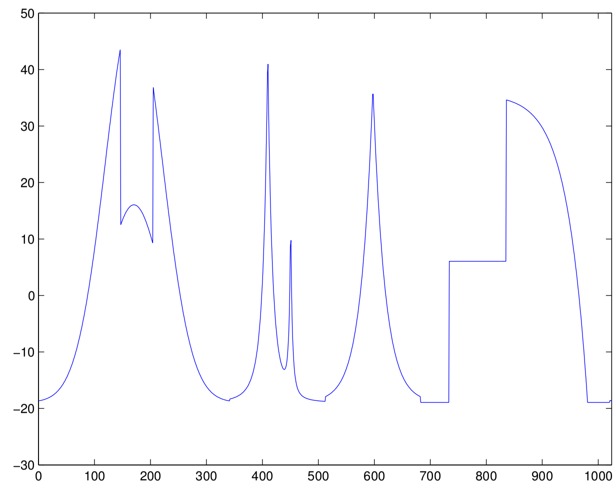


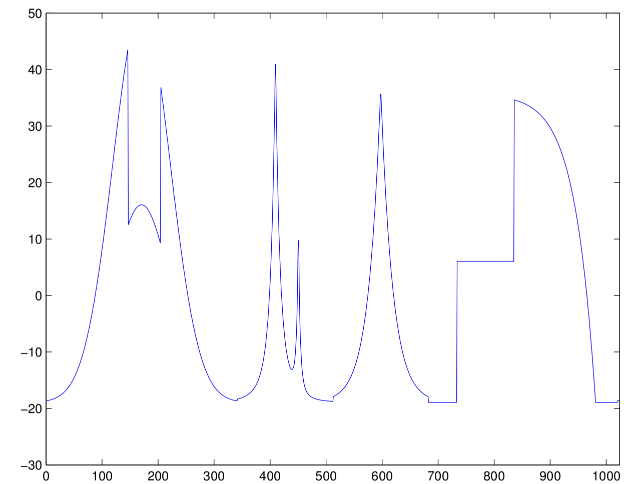
32

Approximation non linéaire

Approximation non linéaire

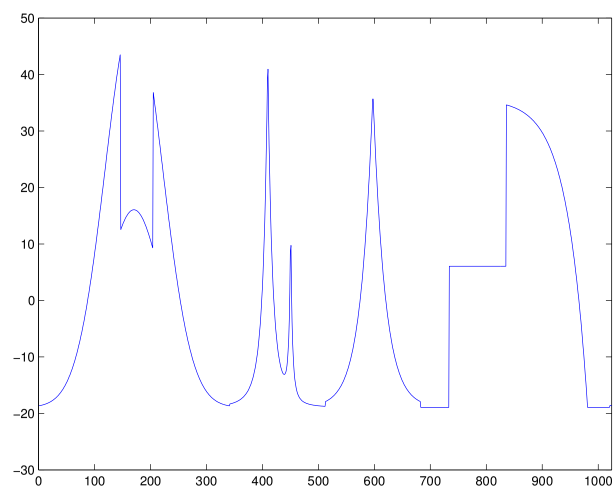
Linéaire

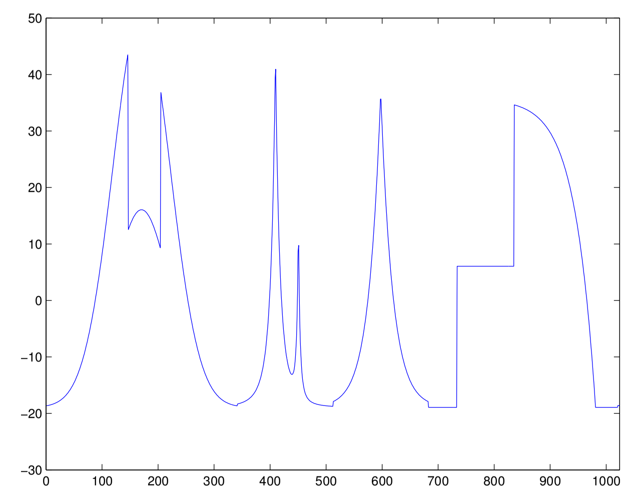




1024 Fourier Ondelettes

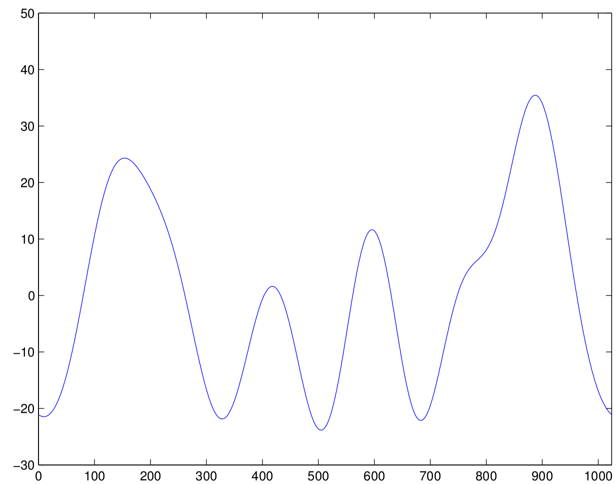
Non Linéaire

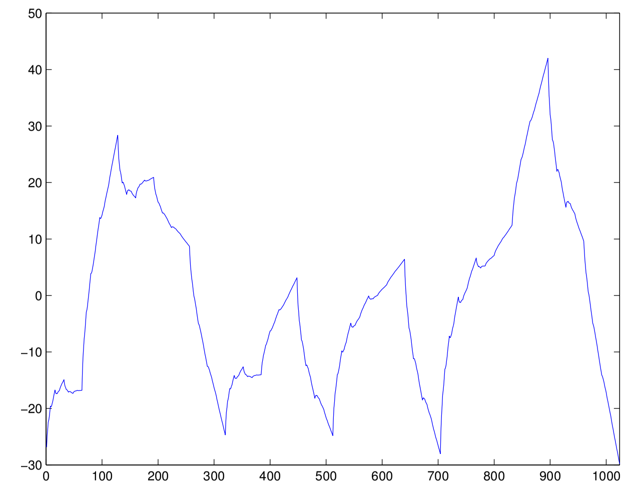




Approximation non linéaire

Linéaire



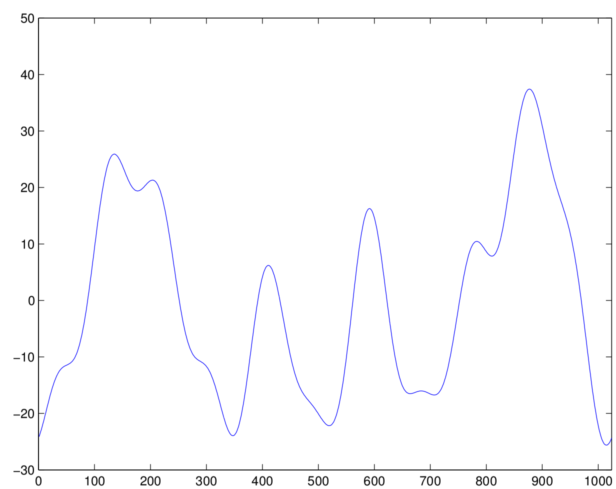


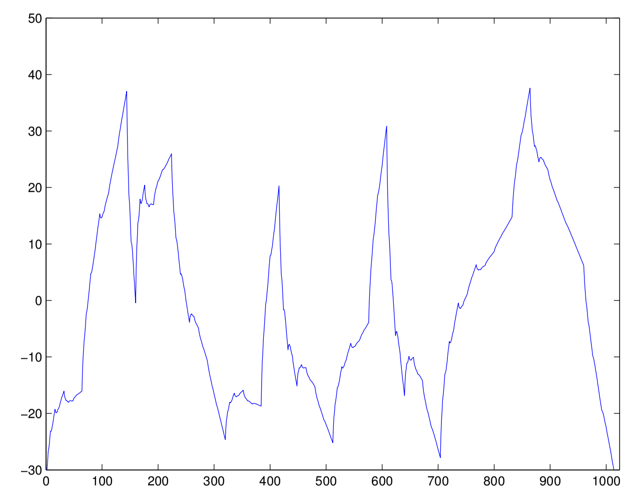
16

Fourier

Ondelettes

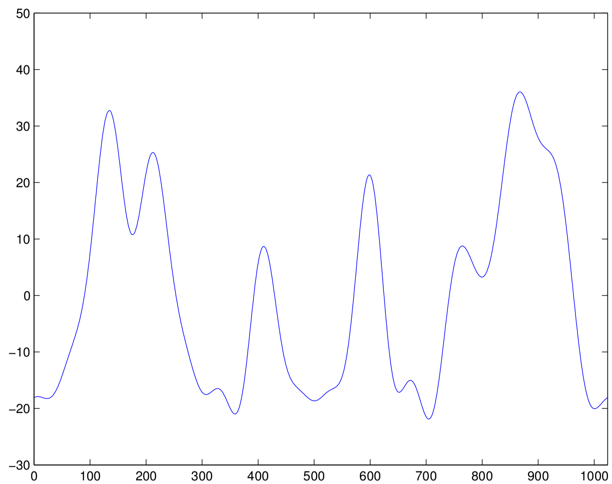
Non Linéaire

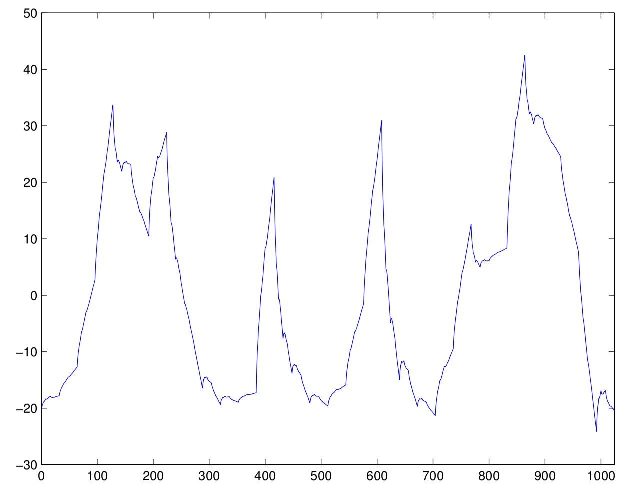




Approximation non linéaire

Linéaire



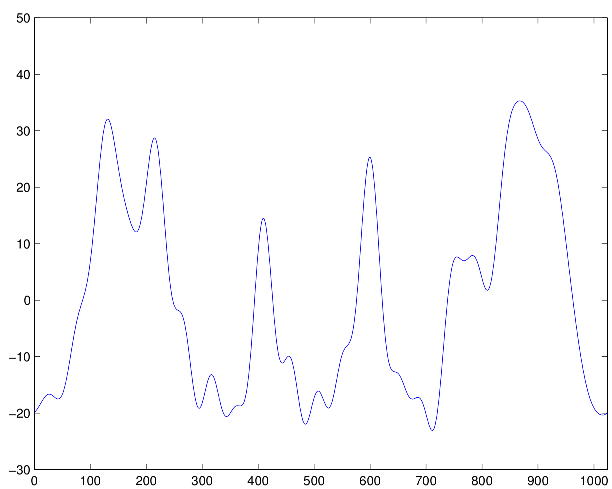


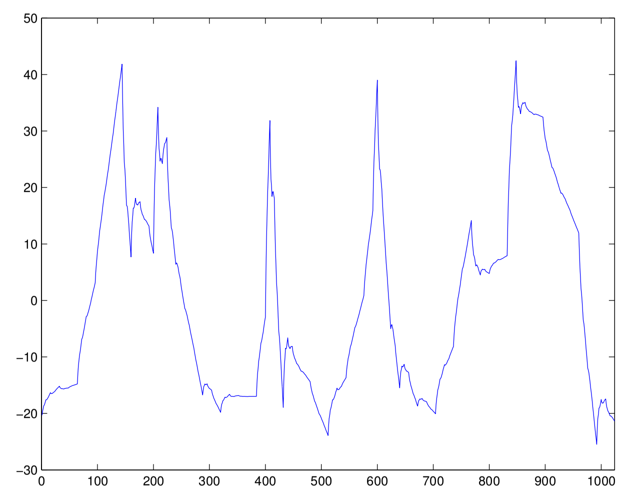
32

Fourier

Ondelettes

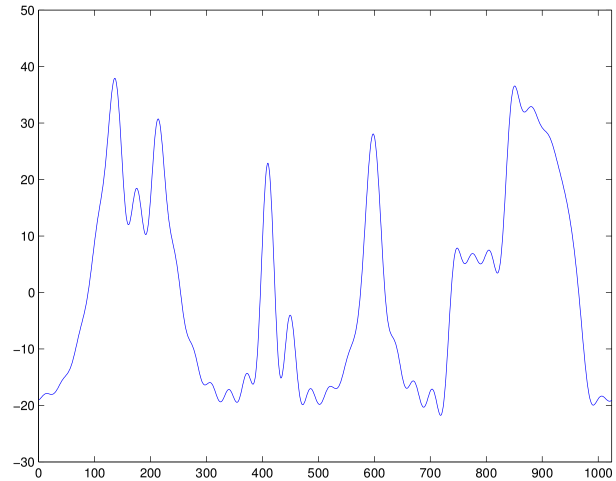
Non Linéaire

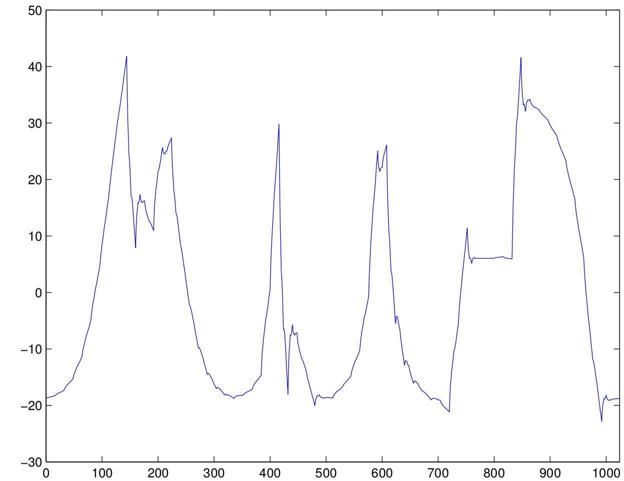




Approximation non linéaire

Linéaire



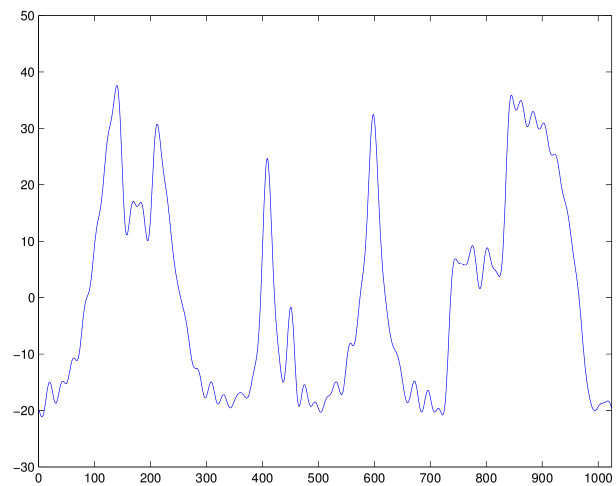


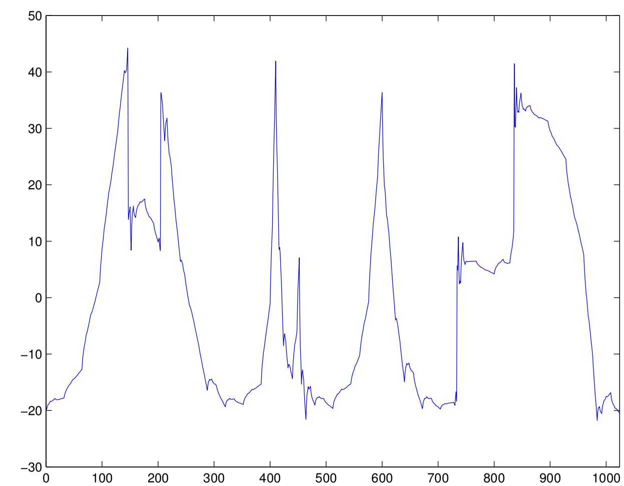
64

Fourier

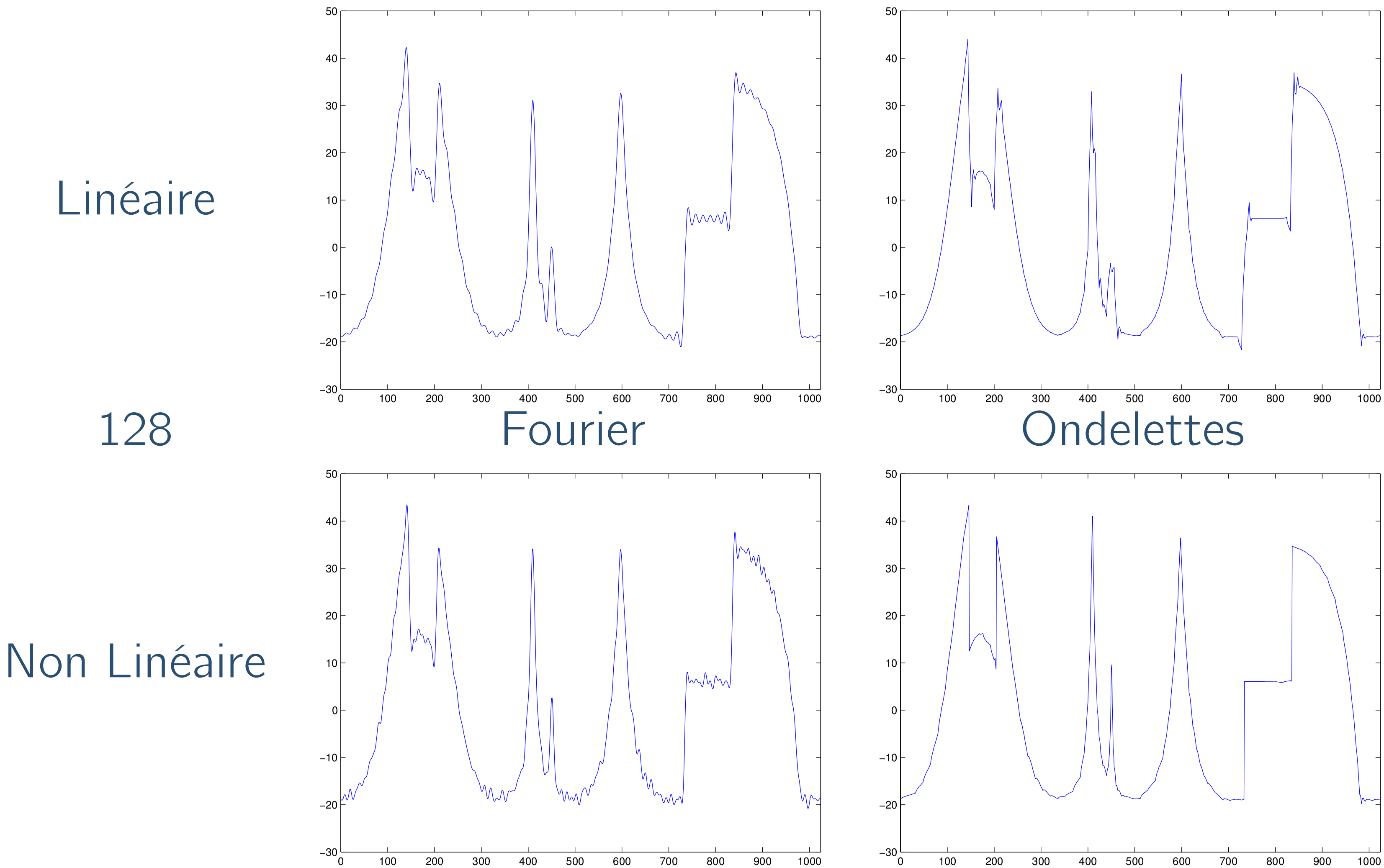
Ondelettes

Non Linéaire



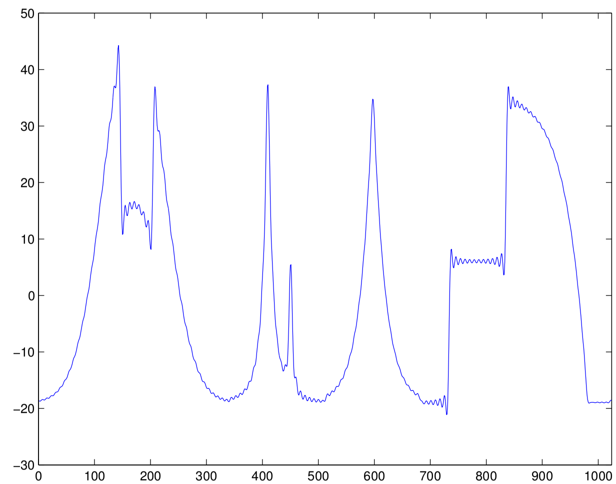


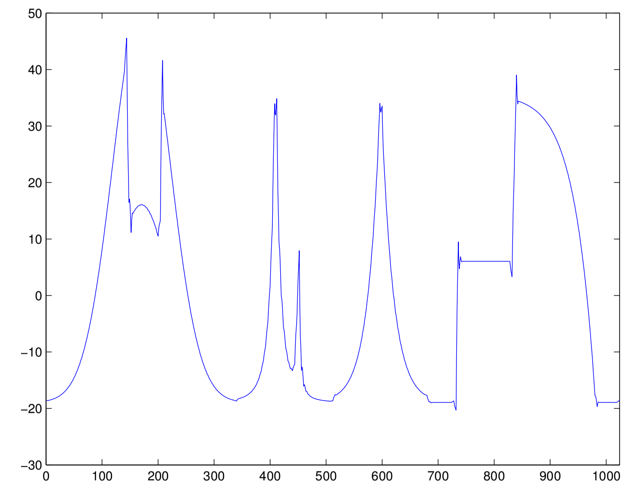
Approximation non linéaire



Approximation non linéaire

Linéaire



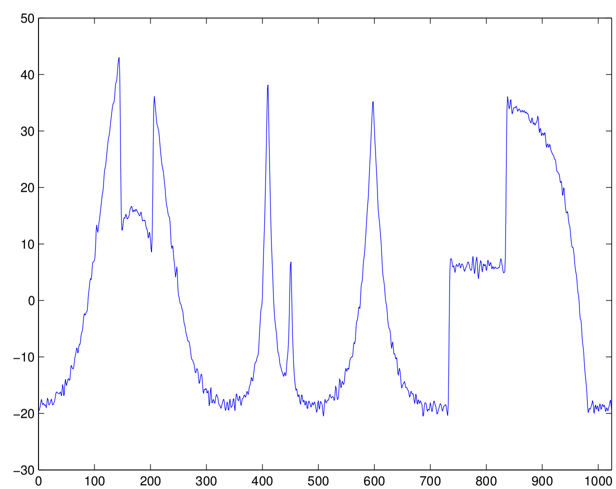


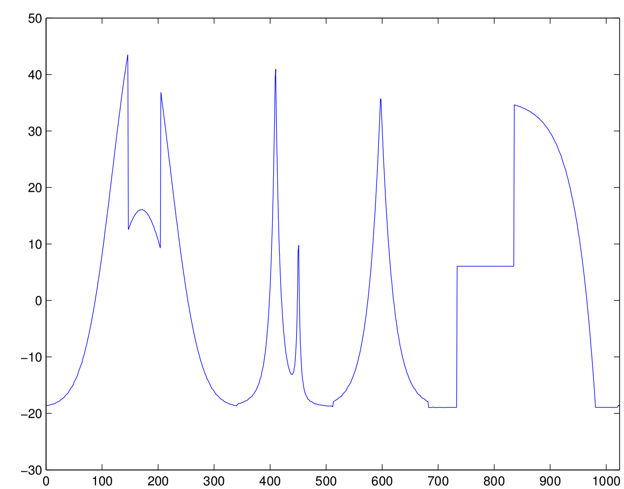
256

Fourier

Ondelettes

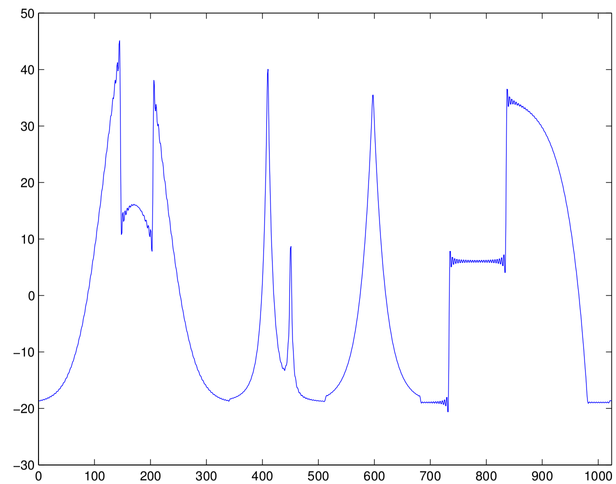
Non Linéaire

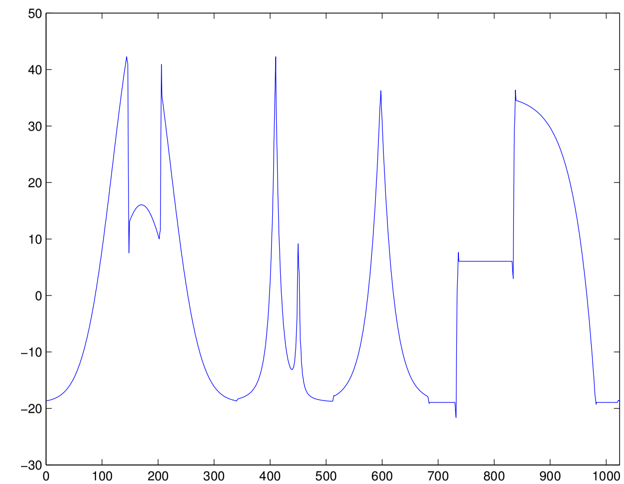




Approximation non linéaire

Linéaire



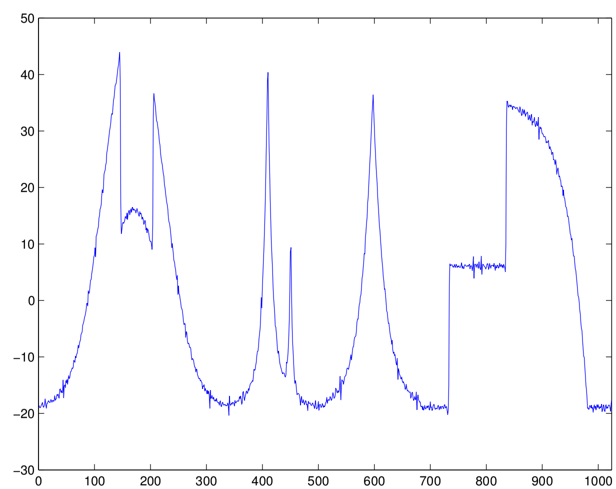


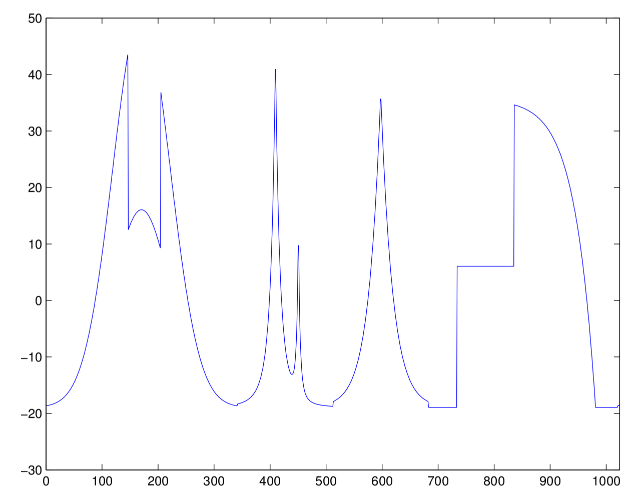
512

Fourier

Ondelettes

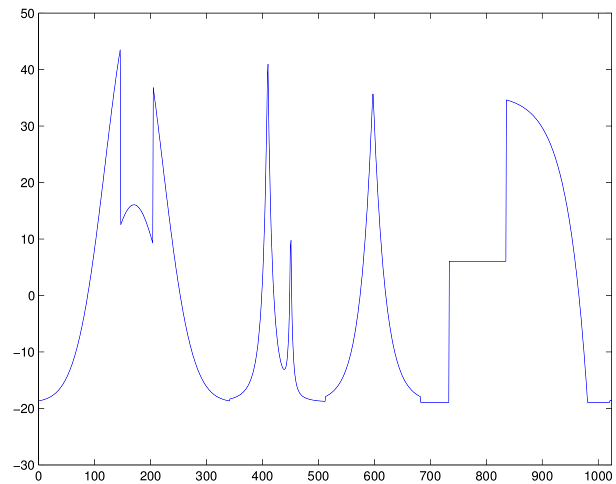
Non Linéaire

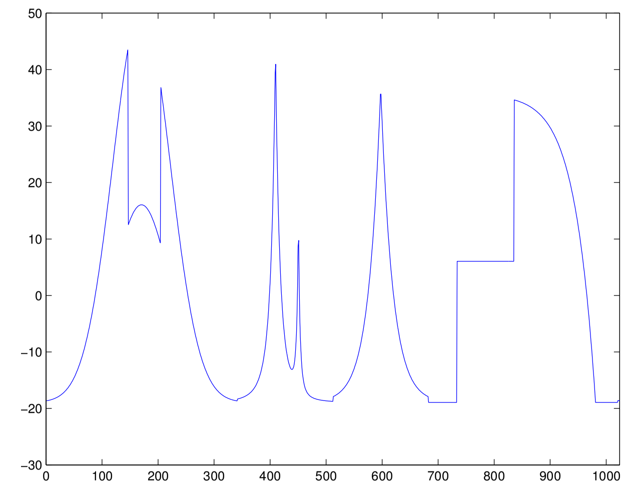




Approximation non linéaire

Linéaire



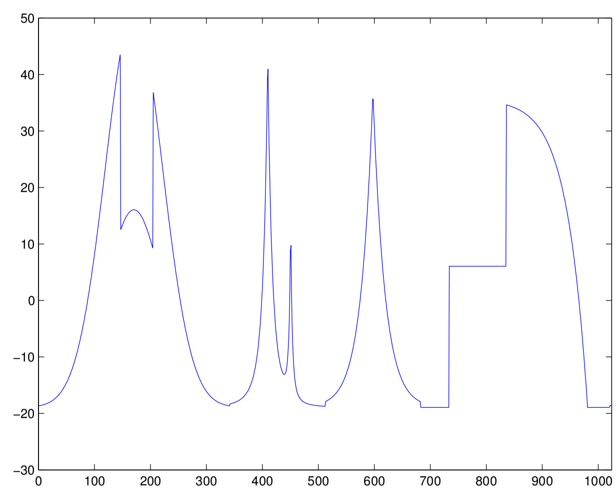


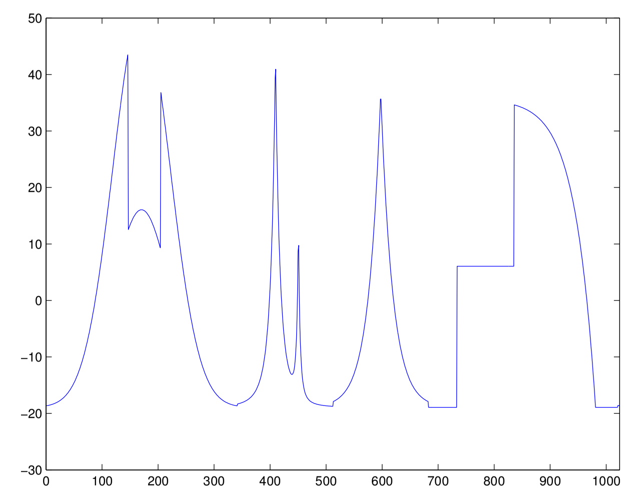
1024

Fourier

Ondelettes

Non Linéaire

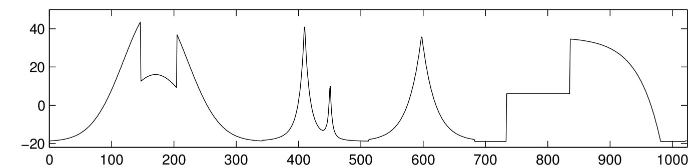




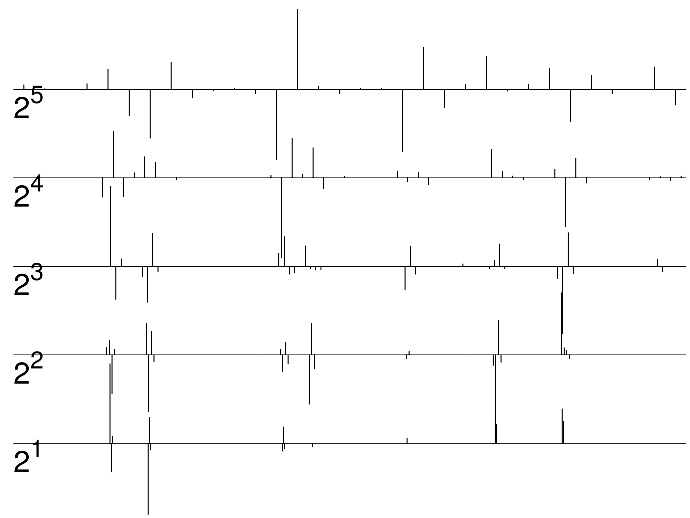
Approximation non linéaire en ondelettes

Approximation non linéaire en ondelettes

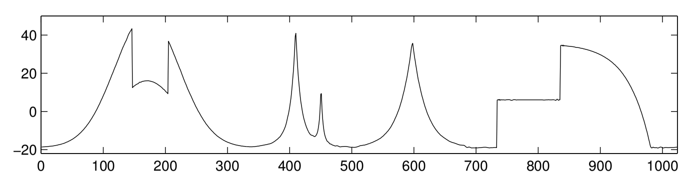
f



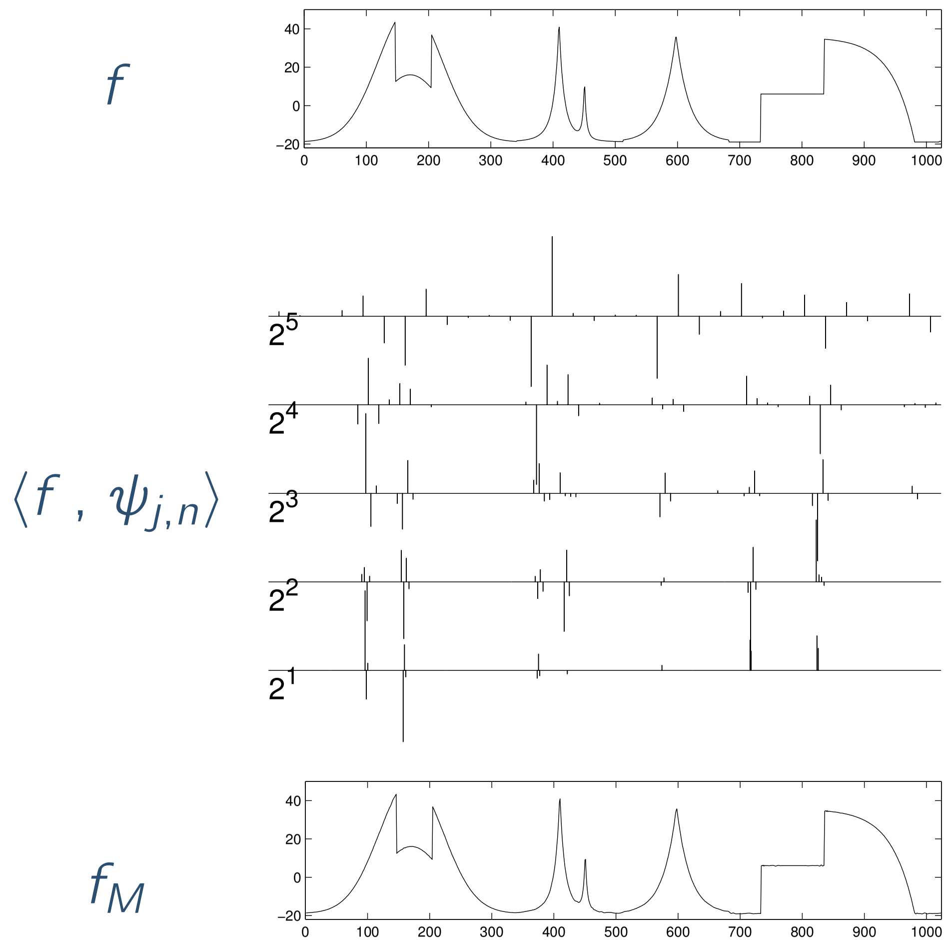
$\langle f\ ,\ \psi_{j,n} \rangle$



f_M



Approximation non linéaire en ondelettes



- Si f est $\mathbf{C}^\alpha$ par morceaux et ψ a $p > \alpha$ moments nuls alors

$$\|f - f_M\|^2 \leq C M^{-2\alpha} \; .$$

Ondelettes 2D

Ondelettes 2D

- Construction similaire mais bidimensionnelle.

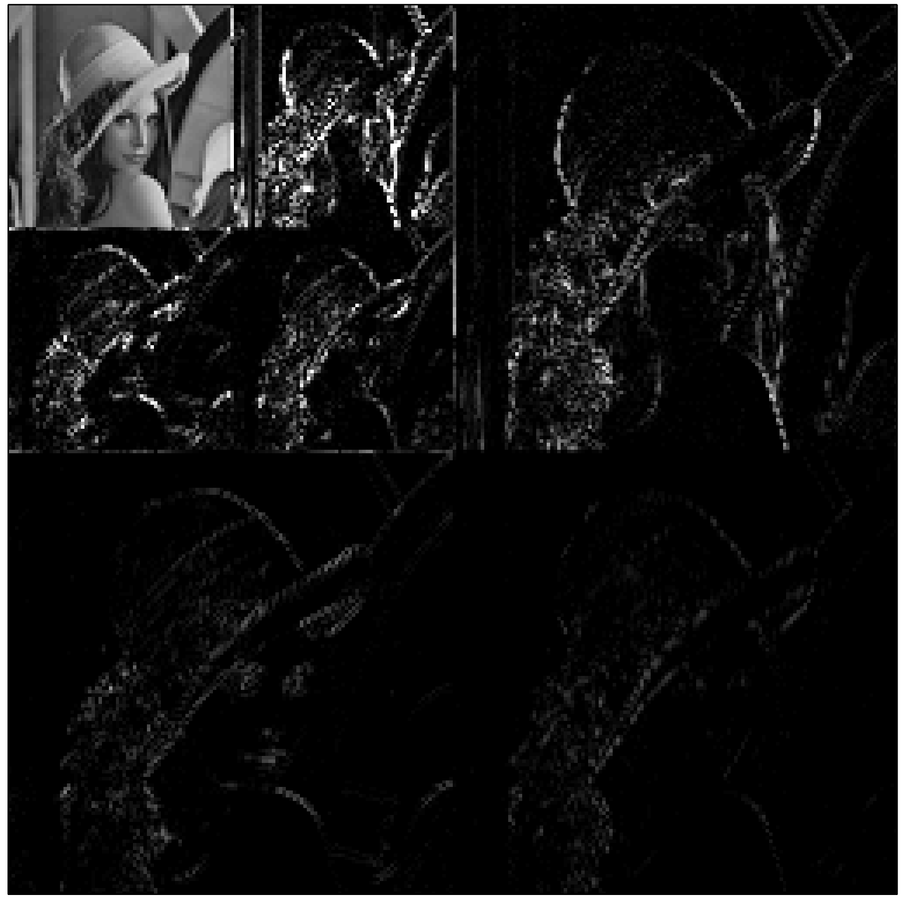
Ondelettes 2D

- Construction similaire mais bidimensionnelle.

Ondelettes 2D

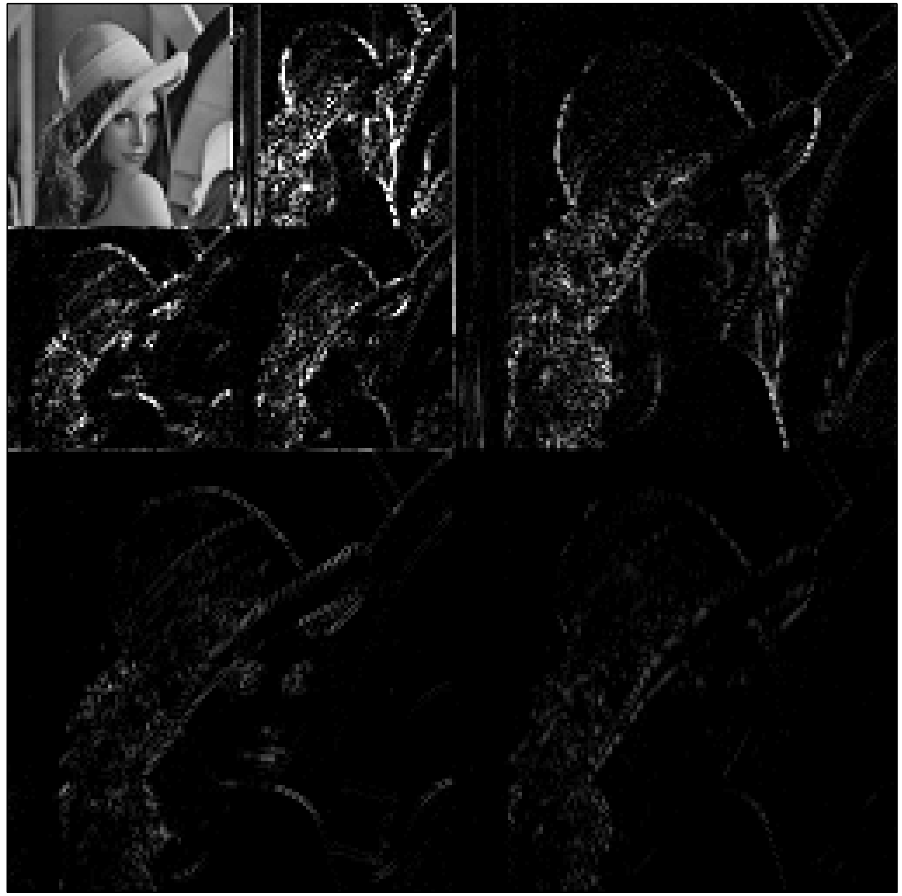
- Construction similaire mais bidimensionnelle.

Ondelettes 2D



- Construction similaire mais bidimensionnelle.

Ondelettes 2D



- Construction similaire mais bidimensionnelle.
- Grands coefficients correspondent aux contours.

Compression par transformée

Compression par transformée

- Décomposition dans une base.

Compression par transformée

- Décomposition dans une base.
- Quantification des coefficients.

Compression par transformée

- Décomposition dans une base.
- Quantification des coefficients.
- Codage entropique.

Compression par transformée

- Décomposition dans une base.
- Quantification des coefficients.
- Codage entropique.
- JPEG (Fourier) utilisé partout (coût/efficacité).

Compression par transformée

- Décomposition dans une base.
- Quantification des coefficients.
- Codage entropique.
- JPEG (Fourier) utilisé partout (coût/efficacité).
- JPEG 2000 (Ondelettes) utilisé dans des applications spécifiques comme Google Earth (besoins spécifiques).

Originale

JPEG

JPEG 2000

Et après

Et après

- Problème clos ?

Et après

- Problème clos ?
- Codage entropique : OUI mais codage source/canal.

Et après

- Problème clos ?
- Codage entropique : OUI mais codage source/canal.
- Modélisation : NON.

Et après

- Problème clos ?
- Codage entropique : OUI mais codage source/canal.
- Modélisation : NON.
- Transformation : NON (théorie de l'approximation).

Et après

- Problème clos ?
- Codage entropique : OUI mais codage source/canal.
- Modélisation : NON.
- Transformation : NON (théorie de l'approximation).
- Vidéo, Sons...

Géométrie

Géométrie

- Caractéristique des images naturelles.

Géométrie

- Caractéristique des images naturelles.
- Inutilisée dans les représentations précédentes.

Géométrie

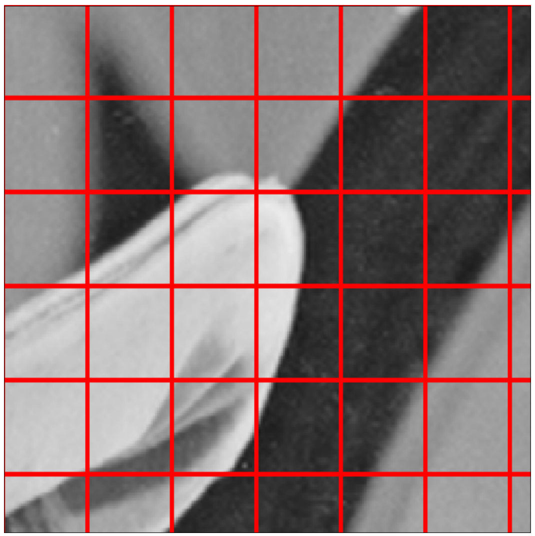
- Caractéristique des images naturelles.
- Inutilisée dans les représentations précédentes.
- Apport théorique prévisible.

Géométrie

- Caractéristique des images naturelles.
- Inutilisée dans les représentations précédentes.
- Apport théorique prévisible.
- Direction de recherche actuelle : curvelets, edgelets, wedgelets, ondelettes géométriques, *bandelettes*. . .

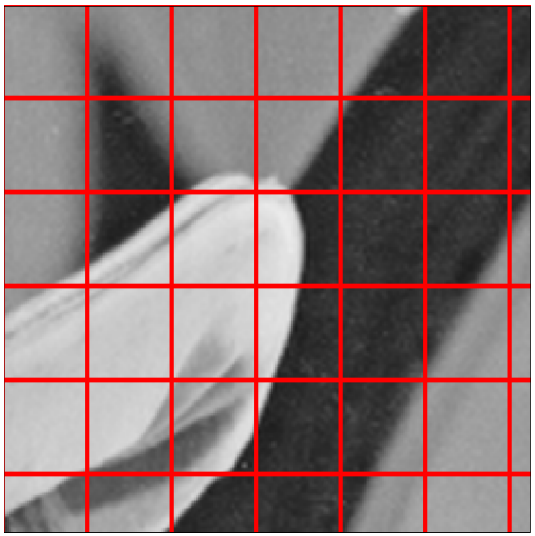
Bandelettes

Bandelettes

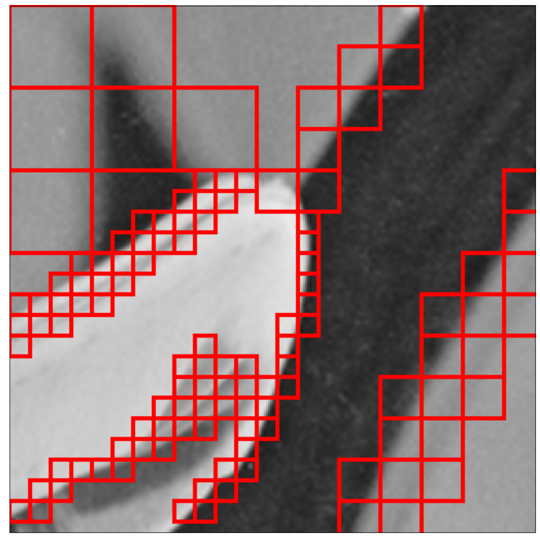


Fourier
Base

Bandelettes

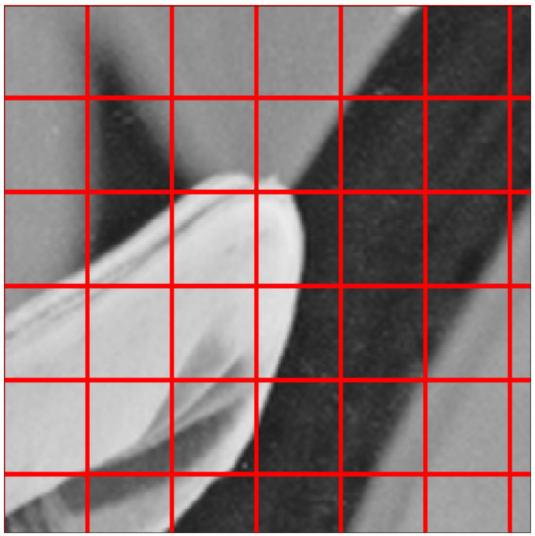


Fourier
Base

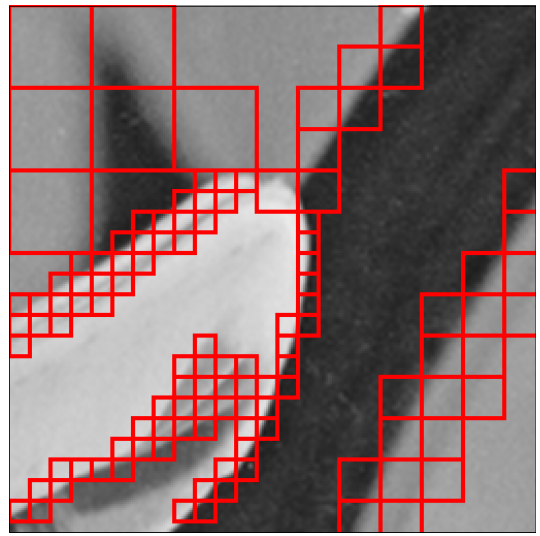


Ondelettes
Multiéchelle

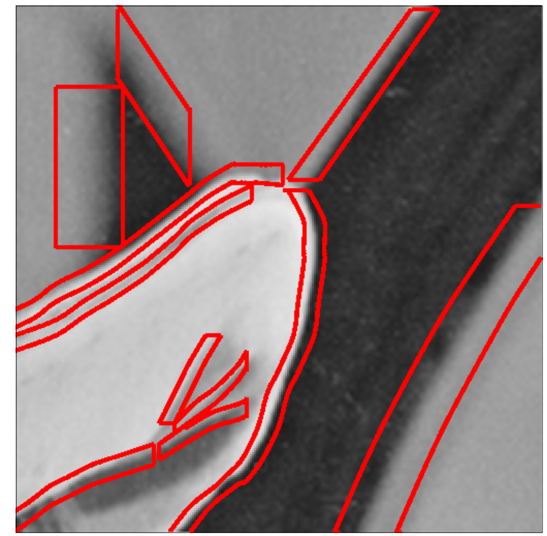
Bandelettes



Fourier
Base

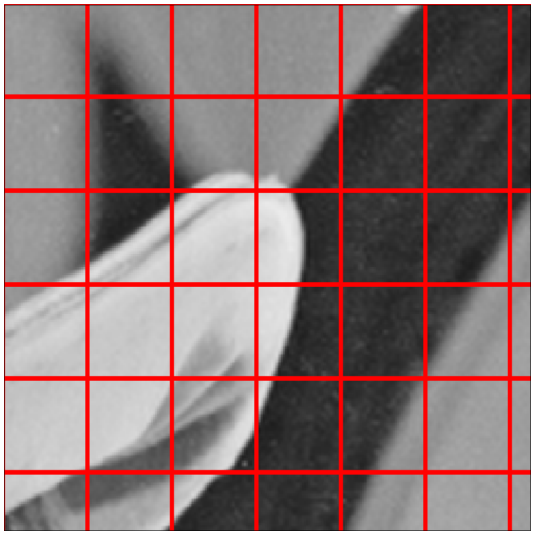


Ondelettes
Multiéchelle

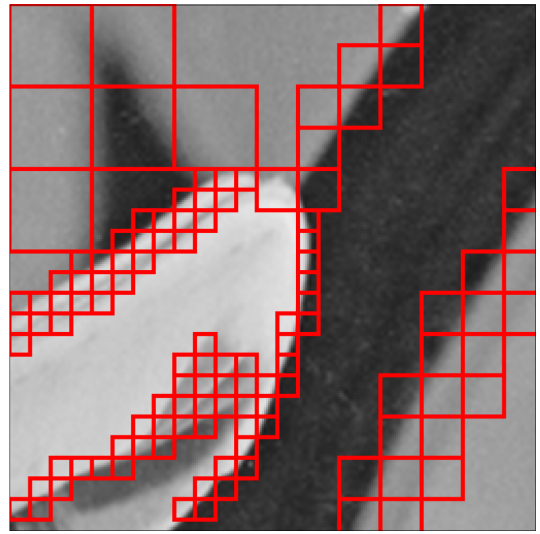


Bandelettes
Géométrie

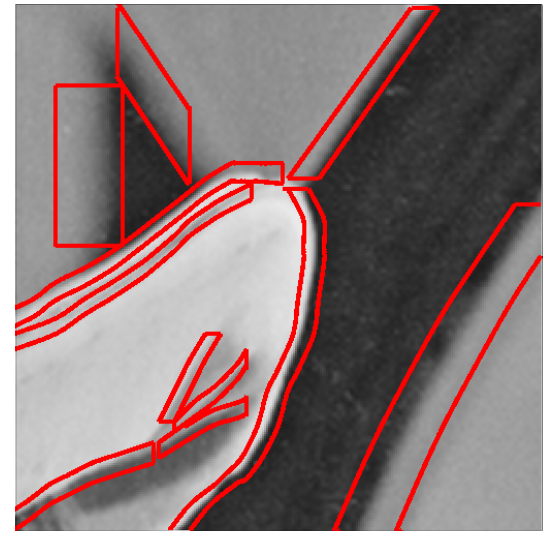
Bandelettes



Fourier
Base



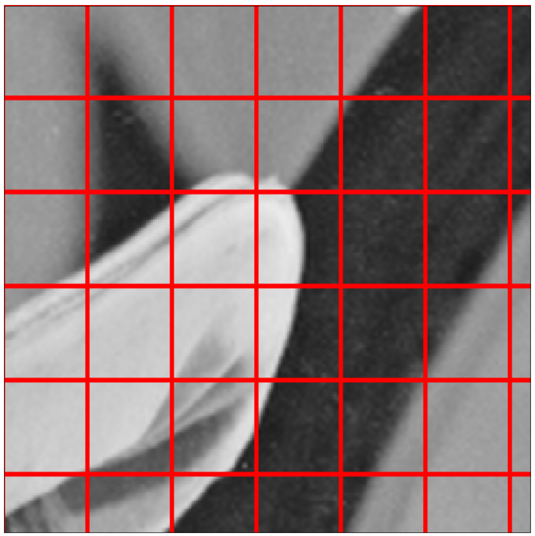
Ondelettes
Multiéchelle



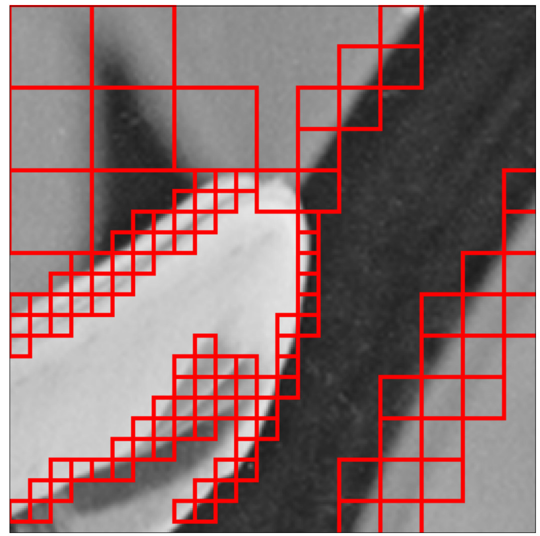
Bandelettes
Géométrie

- Coût : adaptativité, choix de la géométrie.

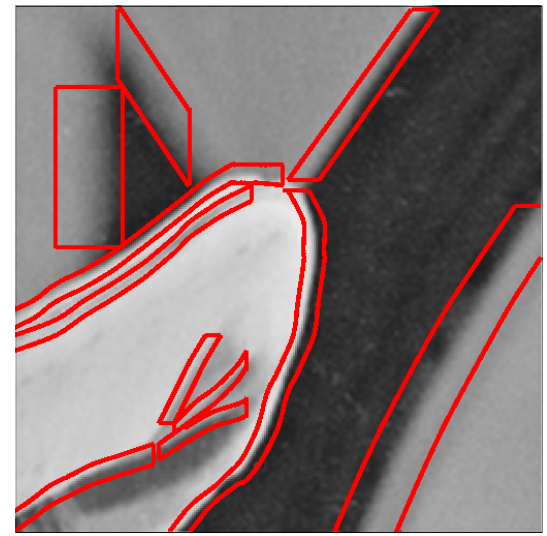
Bandelettes



Fourier
Base



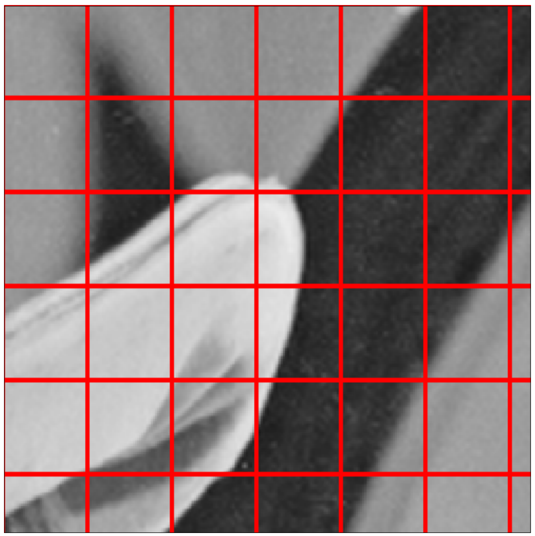
Ondelettes
Multiéchelle



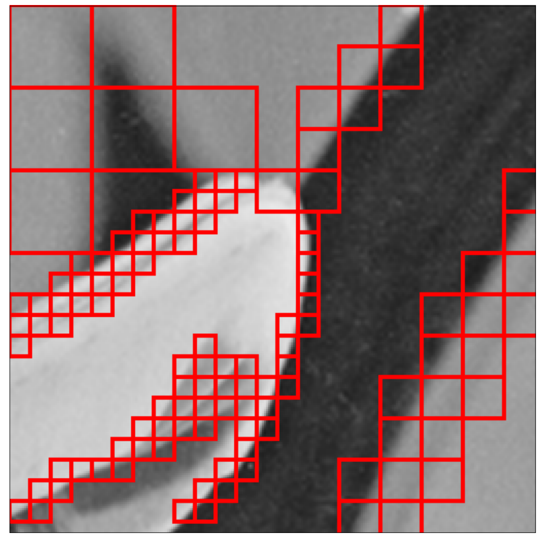
Bandelettes
Géométrie

- Coût : adaptativité, choix de la géométrie.
- Algorithme rapide pour ce choix.

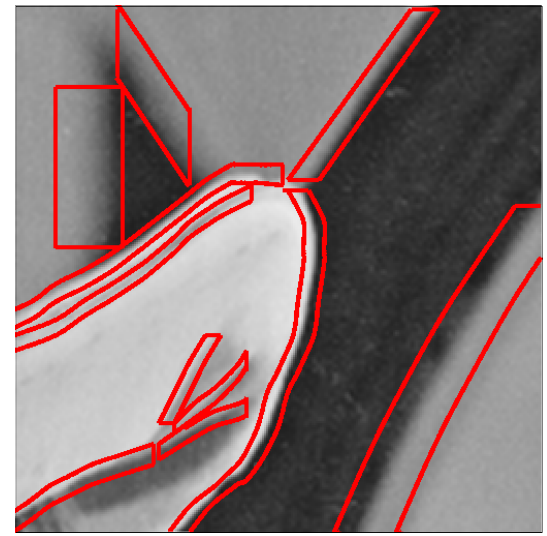
Bandelettes



Fourier
Base



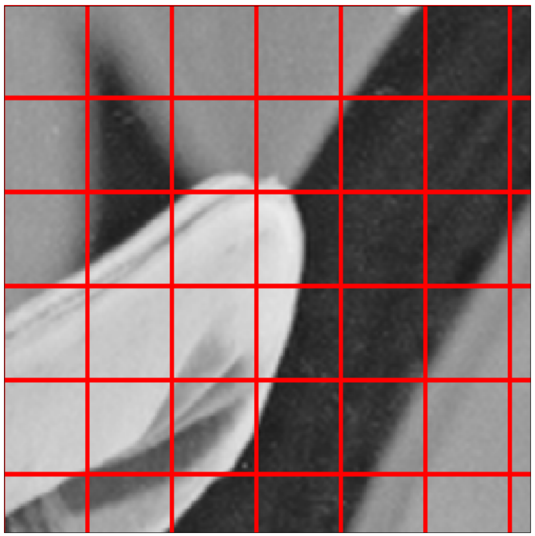
Ondelettes
Multiéchelle



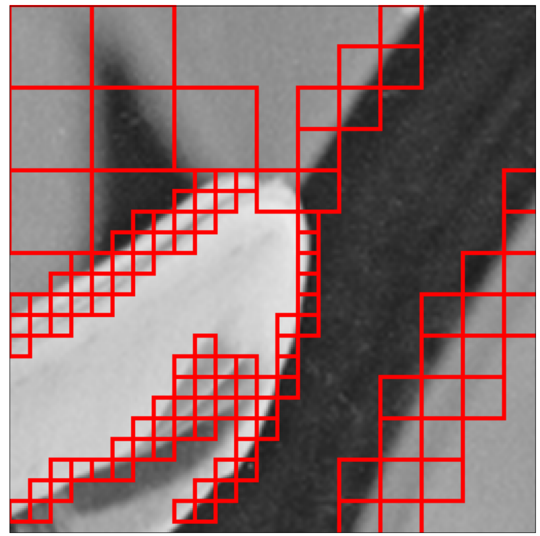
Bandelettes
Géométrie

- Coût : adaptativité, choix de la géométrie.
- Algorithme rapide pour ce choix.
- Travail académique sur l'optimalité de la méthode.

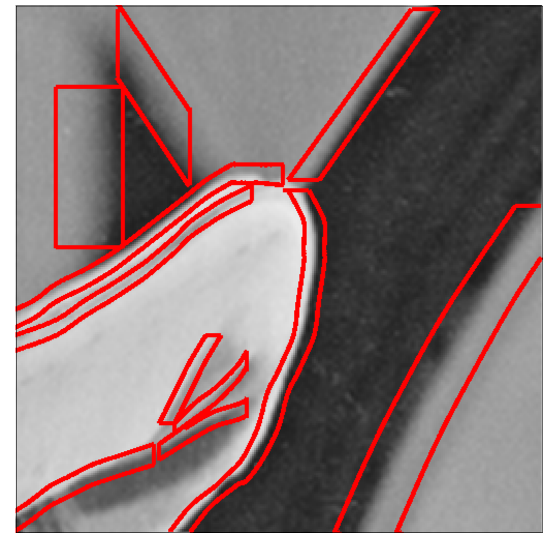
Bandelettes



Fourier
Base



Ondelettes
Multiéchelle

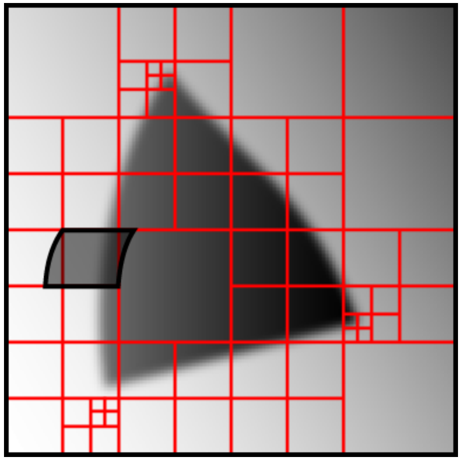


Bandelettes
Géométrie

- Coût : adaptativité, choix de la géométrie.
- Algorithme rapide pour ce choix.
- Travail académique sur l'optimalité de la méthode.
- Implémentation effective sans optimisation fine.

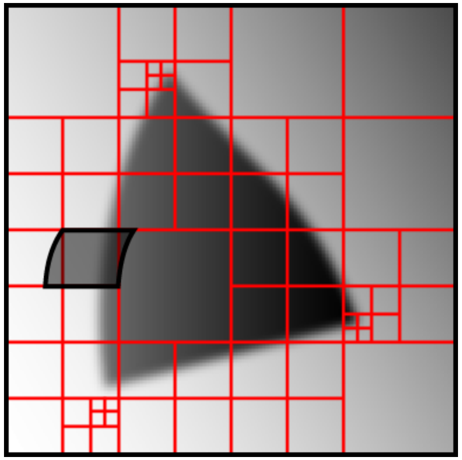
Bandelettes

Bandelettes



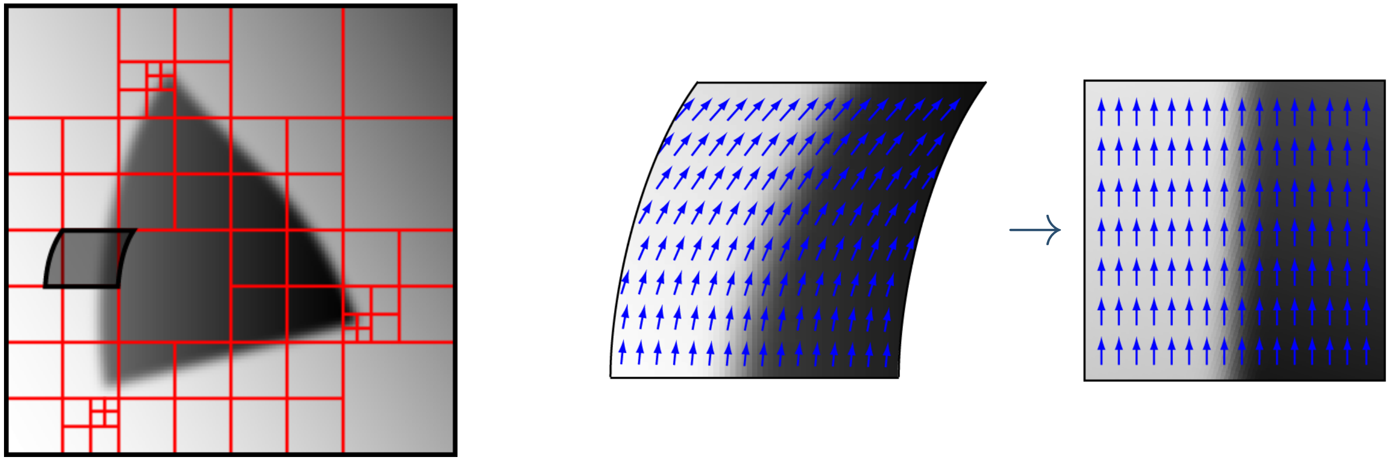
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple

Bandelettes



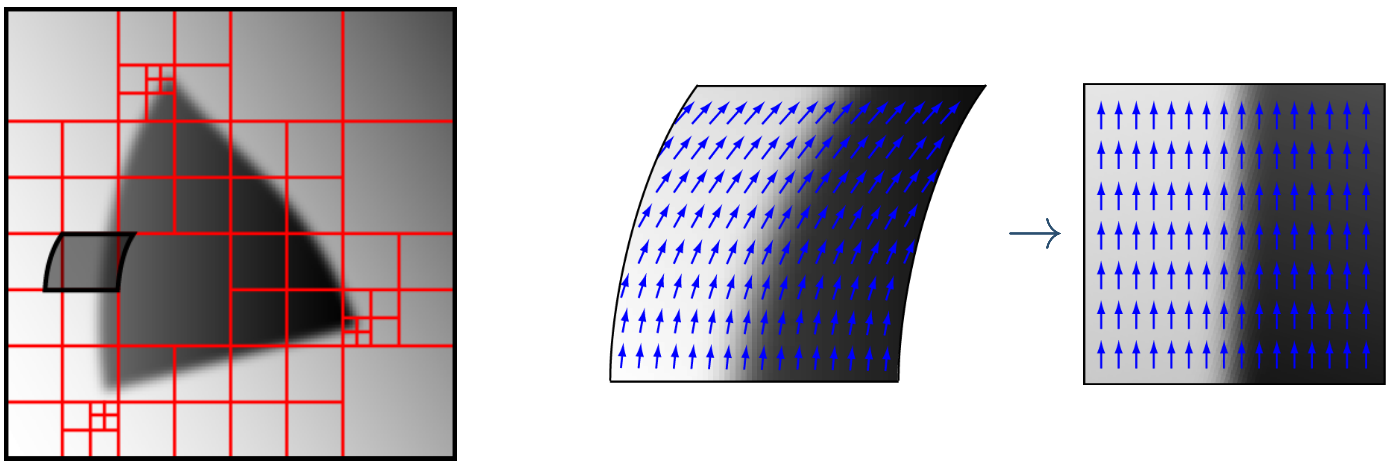
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.

Bandelettes



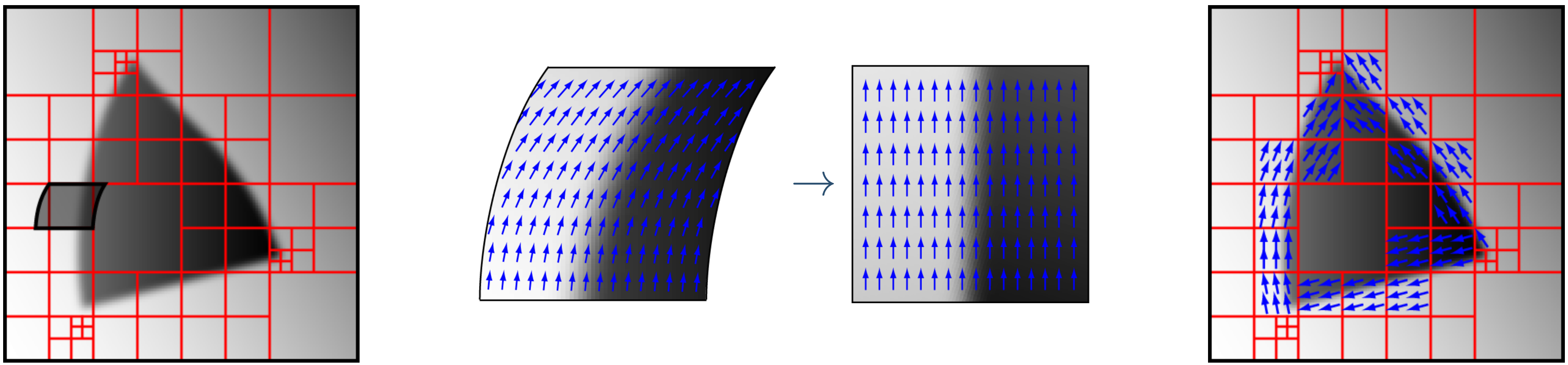
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.

Bandelettes



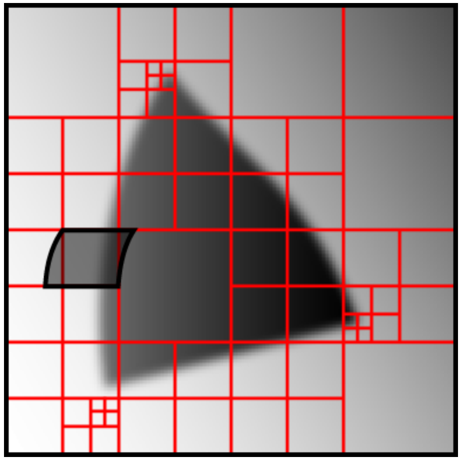
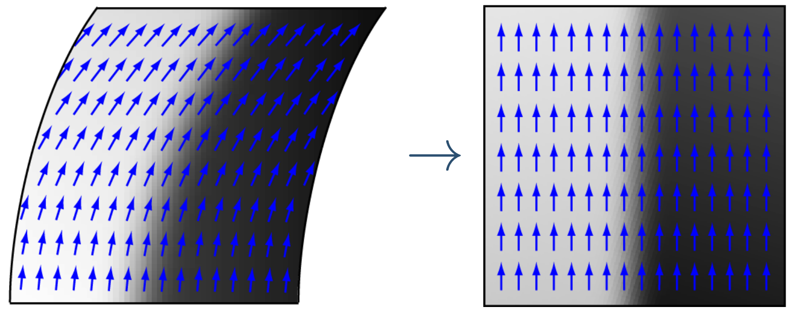
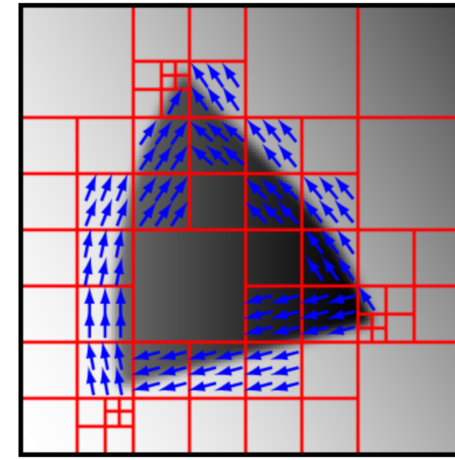
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.

Bandelettes



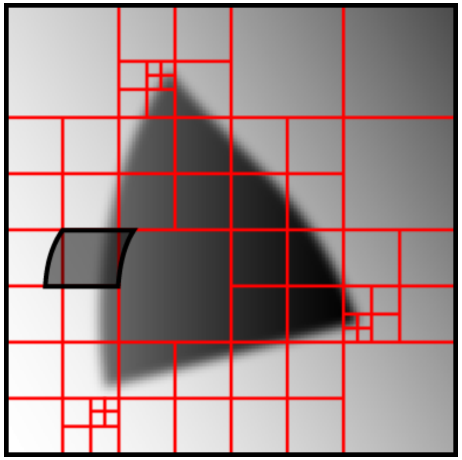
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.
- Base de bandelettes : segmentation dyadique + une géométrie par carré.

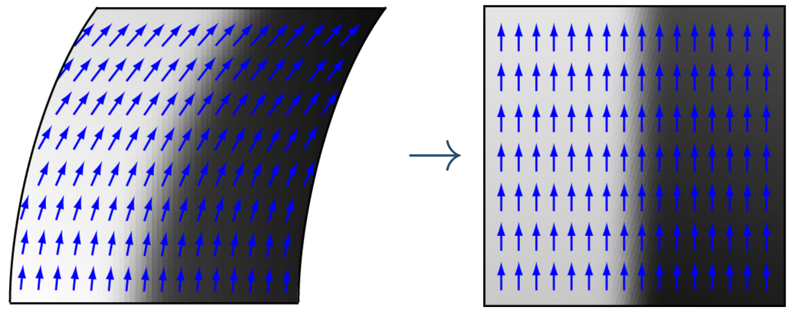
Bandelettes

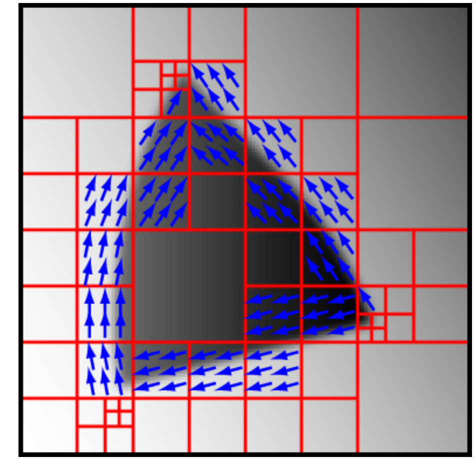




- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.
- Base de bandelettes : segmentation dyadique + une géométrie par carré.
- **Théorème :** Si f est $\mathbf{C}^\alpha - \mathbf{C}^\alpha$, alors, dans une meilleure base,
$$\|f - f_M\|^2 \le C(\log M) M^{-\alpha} \quad .$$

Bandelettes

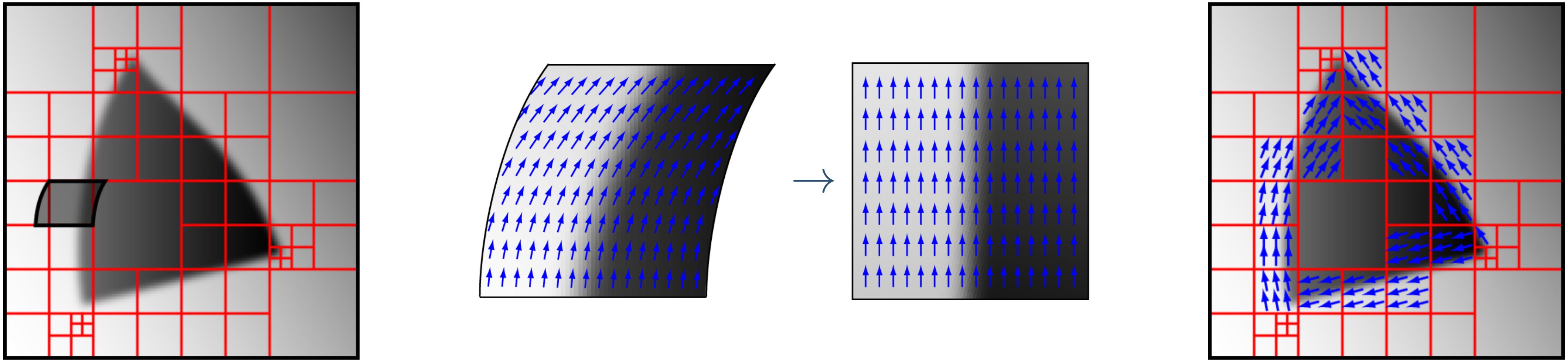






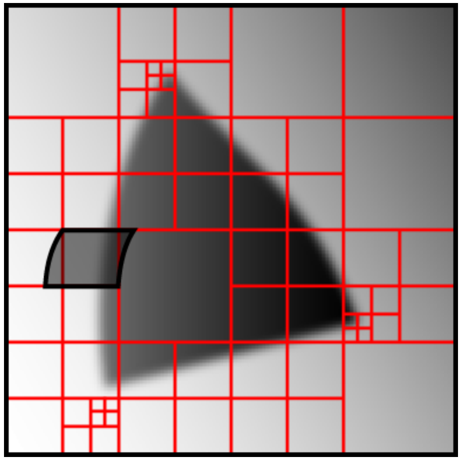
- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.
- Base de bandelettes : segmentation dyadique + une géométrie par carré.
- **Théorème :** Si f est $\mathbf{C}^\alpha - \mathbf{C}^\alpha$, alors, dans une meilleure base,
$$\|f - f_M\|^2 \leq C(\log M) M^{-\alpha} \quad .$$
- Approche Lagrangienne : minimisation de $\|f - f_M\|^2 + T^2 M \Longrightarrow$ seuillage dans une base fixe (facile) et recherche d'une meilleure base (difficile).

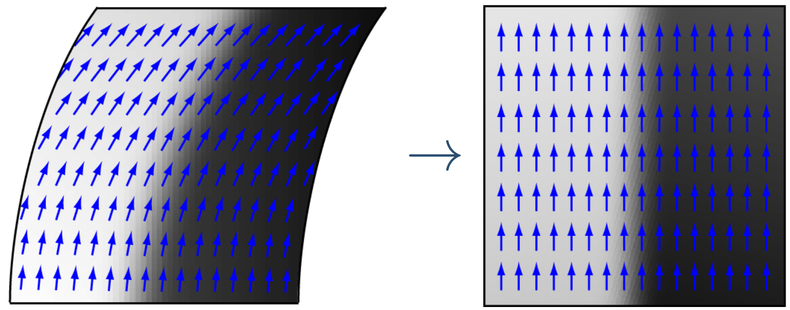
Bandelettes

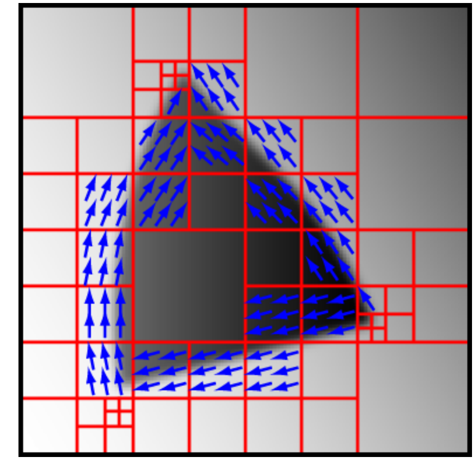


- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.
- Base de bandelettes : segmentation dyadique + une géométrie par carré.
- **Théorème :** Si f est $\mathbf{C}^\alpha - \mathbf{C}^\alpha$, alors, dans une meilleure base,
$$\|f - f_M\|^2 \leq C(\log M) M^{-\alpha} \quad .$$
- Approche Lagrangienne : minimisation de $\|f - f_M\|^2 + T^2 M \Longrightarrow$ seuillage dans une base fixe (facile) et recherche d'une meilleure base (difficile).
- Structure hiérarchique de la segmentation et additivité du Lagrangien : algorithme de meilleure base de Wickerhauser (CART).

Bandelettes



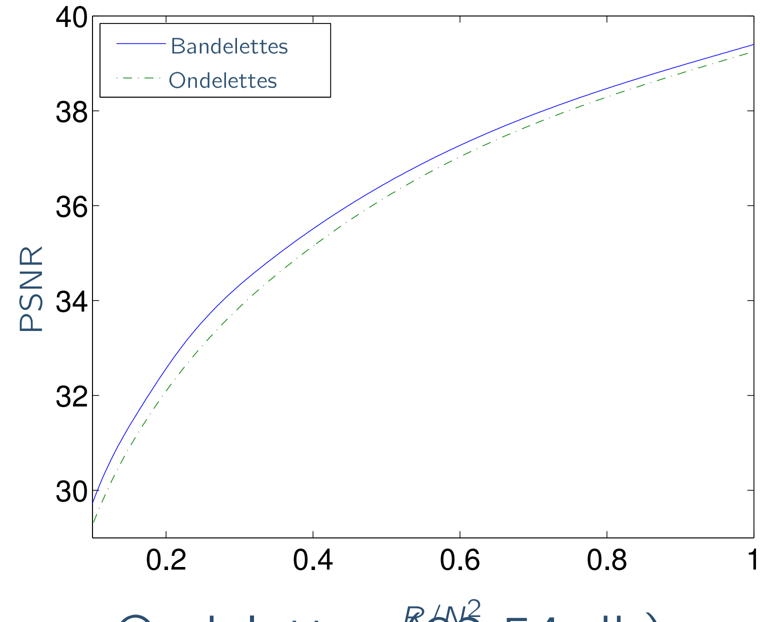




- Image $\mathbf{C}^\alpha - \mathbf{C}^\alpha$ simple par morceaux.
- Déformation locale $\Longrightarrow$ singularité verticale/horizontale.
- Bandelettes locales : préimage d'une base adaptée.
- Base de bandelettes : segmentation dyadique + une géométrie par carré.
- **Théorème :** Si f est $\mathbf{C}^\alpha - \mathbf{C}^\alpha$, alors, dans une meilleure base,
$$\|f - f_M\|^2 \leq C(\log M) M^{-\alpha} \quad .$$
- Approche Lagrangienne : minimisation de $\|f - f_M\|^2 + T^2 M \Longrightarrow$ seuillage dans une base fixe (facile) et recherche d'une meilleure base (difficile).
- Structure hiérarchique de la segmentation et additivité du Lagrangien : algorithme de meilleure base de Wickerhauser (CART).
- Exploration exhaustive des géométries dans chaque carré ($\neq$ détection).

Originale

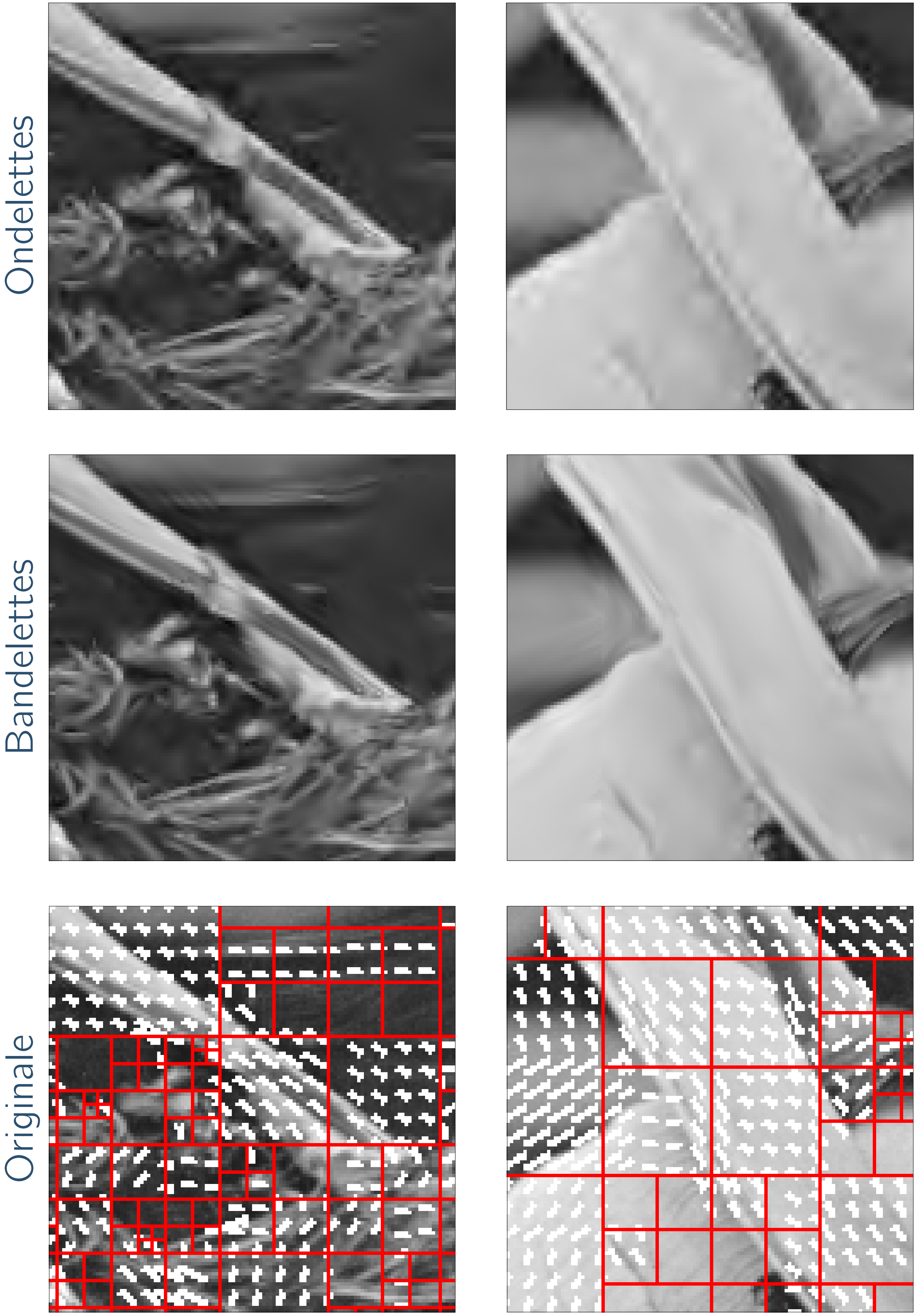
Distorsion-Débit



$R/N^2 = 0{,}22$ bpp

Bandelettes (33,05 db)

Ondelettes (32,54 db)


Ondelettes
Bandelettes
Originale

Modèle spécialisé

Modèle spécialisé

- Compression spécialisée pour les visages.

Modèle spécialisé

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !

Modèle spécialisé

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !
- Comparaison de différents algorithmes.

Modèle spécialisé

JPEG

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !
- Comparaison de différents algorithmes.

Modèle spécialisé

JPEG

JPEG-2000

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !
- Comparaison de différents algorithmes.

Modèle spécialisé

JPEG

JPEG-2000

Let It Wave

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !
- Comparaison de différents algorithmes.

Modèle spécialisé


JPEG

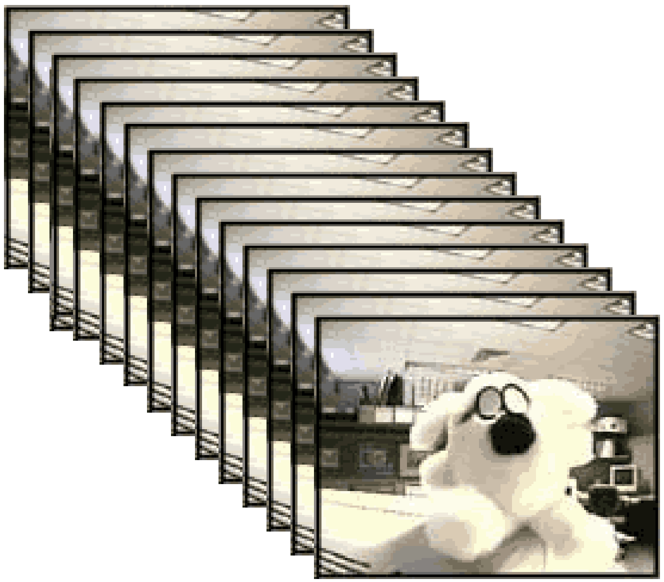

JPEG-2000

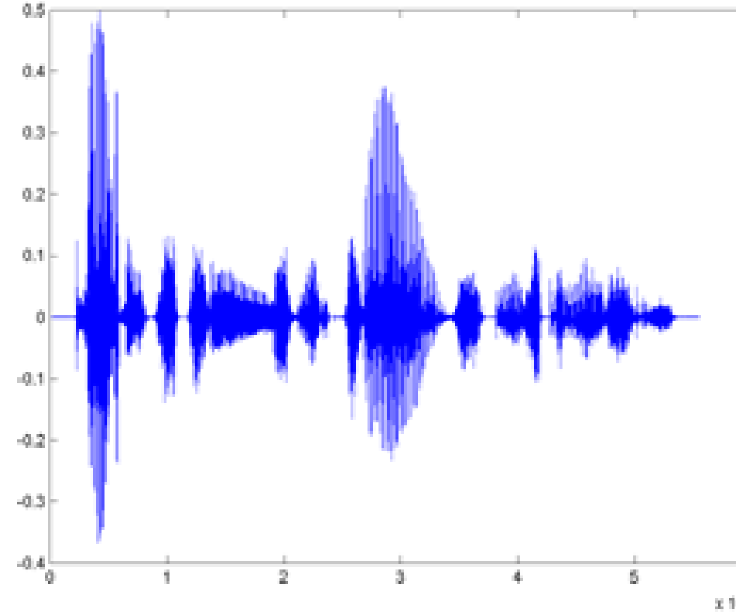

Let It Wave

- Compression spécialisée pour les visages.
- 500 octets = facteur 400 de compression !
- Comparaison de différents algorithmes.
- Clé : spécialisation.

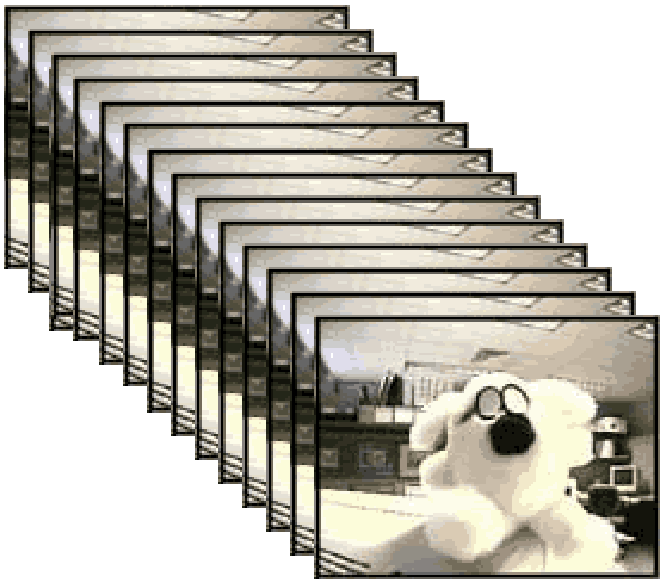
Vidéos et sons

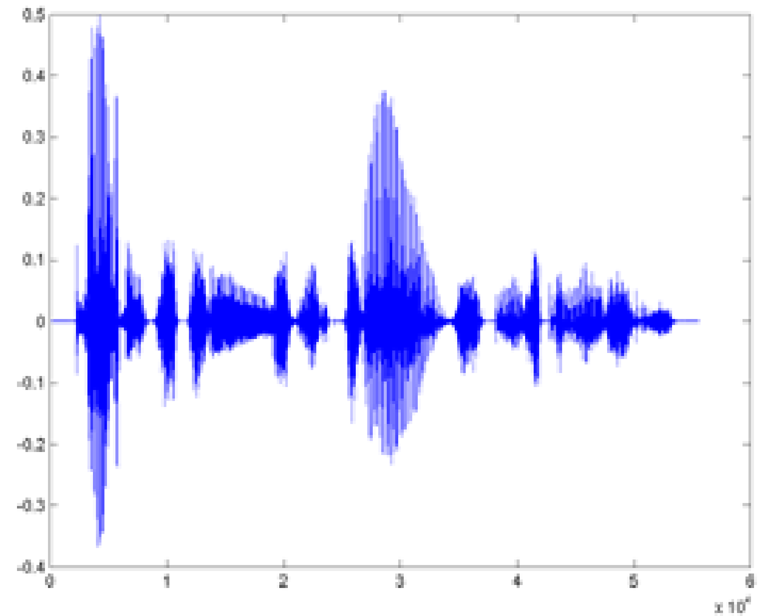
Vidéos et sons





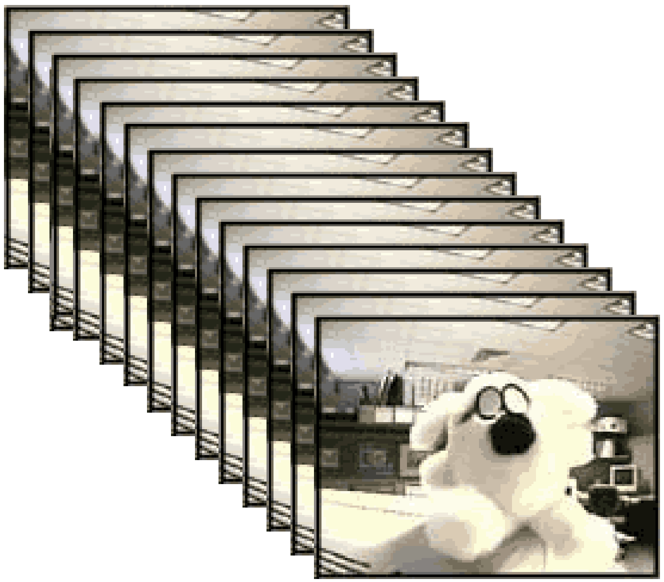
Vidéos et sons

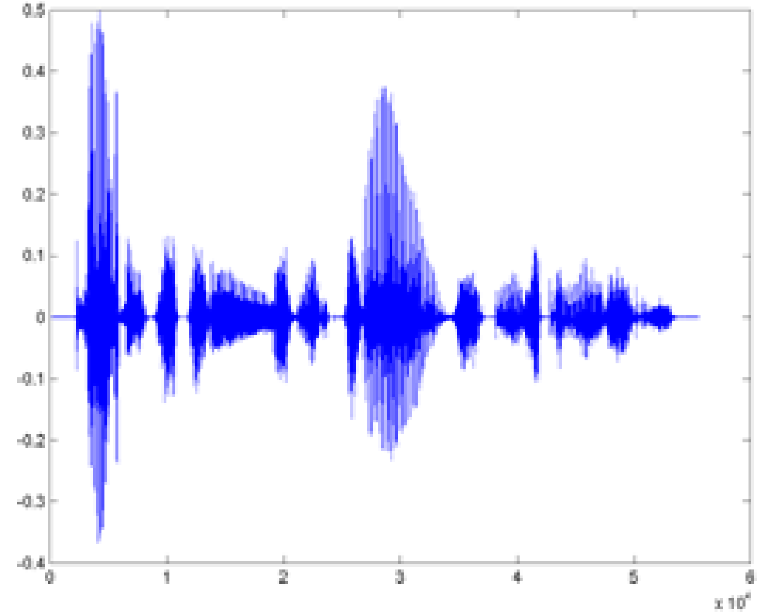




- Mêmes principes s'appliquent !

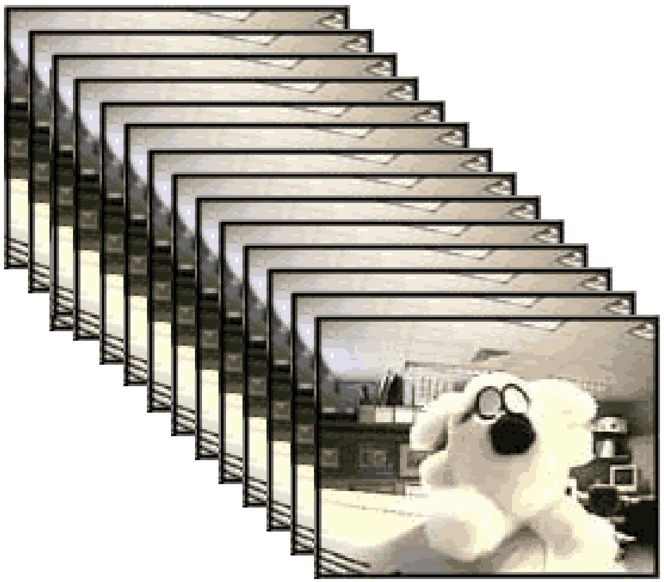
Vidéos et sons

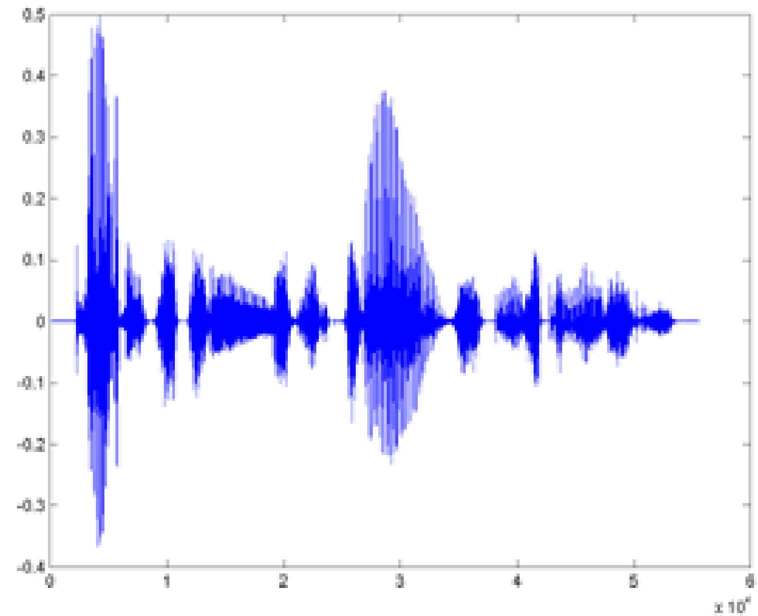




- Mêmes principes s'appliquent !
- Vidéos : utilisation de la redondance temporelle (MPEG2, MPEG4,. . .).

Vidéos et sons





- Mêmes principes s'appliquent !
- Vidéos : utilisation de la redondance temporelle (MPEG2, MPEG4,. . .).
- Sons : utilisation de modèles auditifs (MP3,. . .).

Conclusion

Conclusion

- Survol de la compression d'image.

Conclusion

- Survol de la compression d’image.
- Fondations mathématiques.

Conclusion

- Survol de la compression d’image.
- Fondations mathématiques.
- Théorie de l’information et théorie de l’approximation.

Conclusion

- Survol de la compression d'image.
- Fondations mathématiques.
- Théorie de l'information et théorie de l'approximation.
  item Les mathématiques servent ~~(parfois)~~ à quelque chose !

Conclusion

- Survol de la compression d'image.
- Fondations mathématiques.
- Théorie de l'information et théorie de l'approximation.
  item Les mathématiques servent ~~(parfois)~~ à quelque chose !
- Plus d'infos :
  - `Erwan.Le_Pennec@inria.fr`
  - `http://www.math.jussieu.fr/~lepennec`

Conclusion

- Survol de la compression d'image.
- Fondations mathématiques.
- Théorie de l'information et théorie de l'approximation.
  item Les mathématiques servent ~~(parfois)~~ à quelque chose !
- Plus d'infos :
  - `Erwan.Le_Pennec@inria.fr`
  - `http://www.math.jussieu.fr/~lepennec`
  - En particulier des présentations et un article de vulgarisation sur la compression.