

TP10 – Décomposition *cartoon+texture* d'une image

Une des multiples applications de la transformation de Fourier (TF) est le *filtrage spectral*. Soit un signal unidimensionnel $f(t) : \mathbb{R} \rightarrow \mathbb{R}$. On appelle *filtre spectral* toute application qui, au *spectre d'entrée* $F_e(\nu) : \mathbb{R} \rightarrow \mathbb{C}$ de $f(t)$, obtenu par TF, associe un *spectre de sortie* $F_s(\nu) : \mathbb{R} \rightarrow \mathbb{C}$ tel que $\forall \nu \in \mathbb{R}, |F_s(\nu)| \leq |F_e(\nu)|$. Les filtres les plus connus sont les *passes-bas* et les *passes-haut*. Les filtres des figures 1-a et 1-b, qui sont caractérisés par la même *fréquence de coupure*, sont complémentaires : la somme des deux spectres de sortie est égale au spectre d'entrée. On peut construire des filtres plus sophistiqués en combinant judicieusement un filtre passe-bas et un filtre passe-haut de fréquences de coupure différentes : un filtre *passes-bande* s'obtient en faisant subir au spectre d'entrée un filtrage passe-bas puis un filtrage passe-haut (cf. figure 1-c) ; un filtre *coupe-bande* s'obtient en sommant les spectres de sortie résultant d'un filtrage passe-bas et d'un filtrage passe-haut (cf. figure 1-d).

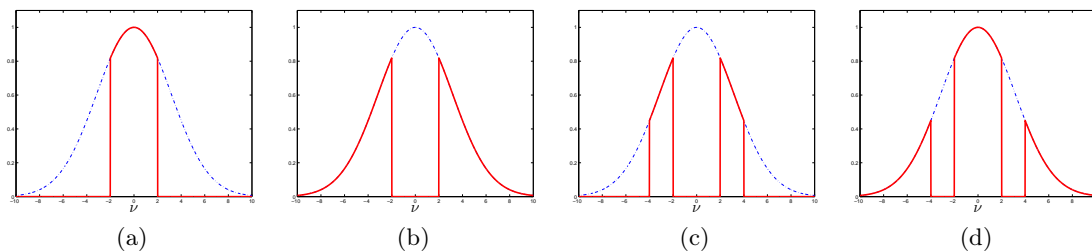


FIGURE 1 – (a) Filtre passe-bas. (b) Filtre passe-haut. (c) Filtre passe-bande. (d) Filtre coupe-bande.

Exercice 1 : partition du spectre d'une image

Complétez le script `exercice_1.m` qui lit une image, calcule son spectre, c'est-à-dire sa transformée de Fourier discrète bidimensionnelle (fonction `fft2` de Matlab), puis décompose ce spectre en deux parties, grâce à un filtre passe-bas Φ_b et un filtre passe-haut Φ_h complémentaires :

$$\begin{cases} \Phi_b(\nu_x, \nu_y) = \chi(\sqrt{\nu_x^2 + \nu_y^2} \leq \nu_c) \\ \Phi_h(\nu_x, \nu_y) = \chi(\sqrt{\nu_x^2 + \nu_y^2} > \nu_c) \end{cases}$$

où $\chi(s)$ vaut 1 si s est vrai et 0 sinon (fonction indicatrice), et où ν_c constitue la fréquence de coupure. Les images correspondant aux basses fréquences et aux hautes fréquences sont obtenues par transformation de Fourier discrète inverse bidimensionnelle (fonction `ifft2` de Matlab), dont on ne garde que la partie réelle.

Ce script permet d'effectuer une séparation grossière de l'image en une partie « plutôt photométrique » et une partie « plutôt géométrique », connue sous le nom de *décomposition cartoon+texture* d'une image. Testez différentes valeurs de ν_c afin d'obtenir une décomposition « optimale » (le critère d'optimalité est très subjectif).

Exercice 2 : filtrage spectral

Même en choisissant la « meilleure » valeur de ν_c , le résultat du script `exercice_1.m` reste visuellement médiocre. Cela vient de la coupure franche entre basses et hautes fréquences. Pour améliorer l'extraction des basses fréquences, une solution consiste à pondérer le spectre non pas par une fonction indicatrice, mais par une fonction lisse décroissant avec la distance à l'origine dans le plan de Fourier.

Faites une copie de `exercice_1.m`, de nom `exercice_2.m`, afin d'utiliser le filtre passe-bas suivant :

$$\Phi_b(\nu_x, \nu_y) = \frac{\eta}{\nu_x^2 + \nu_y^2 + \eta}$$

où le rôle de η est d'éviter la division par 0 lorsque $\nu_x = \nu_y = 0$ (la valeur de η pourra être fixée à 0.05).

Exercice 3 : modèle ROF

Le problème de la décomposition *cartoon+texture* peut être résolu sans passer explicitement dans le domaine de Fourier. La décomposition d'une image I en deux parties C (*cartoon*, qui correspond aux basses fréquences) et T (*texture*, qui correspond aux hautes fréquences) est telle que $C + T = I$, si les filtres passe-bas et passe-haut utilisés sont complémentaires, car la transformation de Fourier est linéaire. L'image T se déduit donc très simplement de l'image C . Par ailleurs, nous attendons de C qu'elle soit « proche » de l'image originale I et que les variations de niveau de gris soient moins brutales dans C que dans I . Nous pouvons donc utiliser le *modèle ROF* (proposé en 1992 par Rudin, Osher et Fatemi), qui fait apparaître la *variation totale* :

$$E_{\text{ROF}}(C) = \iint \left[\frac{\lambda}{2} \|C(x, y) - I(x, y)\|^2 + \|\nabla C(x, y)\|_\epsilon \right] dx dy \quad (1)$$

où $\lambda > 0$ est un paramètre fixé par l'utilisateur et $\|\nabla C(x, y)\|_\epsilon = \sqrt{\|\nabla C(x, y)\|^2 + \epsilon}$. Un modèle très similaire a déjà été rencontré lors du TP8 sur la restauration d'images. D'après le calcul des variations, l'équation d'Euler-Lagrange associée à cette fonctionnelle s'écrit :

$$\lambda [C(x, y) - I(x, y)] - \text{Div} \left(\frac{\nabla C(x, y)}{\|\nabla C(x, y)\|_\epsilon} \right) = 0 \quad (2)$$

L'équation (2) peut être résolue itérativement, étant donné une configuration initiale de C que l'on choisit évidemment égale à I , par un schéma en tout point identique à celui de l'exercice 2 du TP8.

Complétez le script `exercice_3.m` de façon à implémenter ce schéma, en vous inspirant du TP8. Faites-en une copie, de nom `exercice_3_bis.m`, que vous modifierez de manière à lire l'image `empreinte.png`. En seuillant l'image T obtenue au fil des itérations (la valeur du seuil peut être fixée à 20), vous obtiendrez une image binaire de l'empreinte digitale débarrassée des variations progressives du niveau de gris. Pour vous convaincre de l'intérêt de cette méthode d'extraction de la texture, seuillez l'image originale I et comparez les résultats.

Exercice 4 : modèle TV-Hilbert (exercice facultatif)

Il est également possible de résoudre le problème de décomposition *cartoon+texture* d'une image en utilisant le *modèle TV-Hilbert*, qui est un modèle « mixte », car un des deux termes dépend du spectre \hat{C} de l'inconnue C et du spectre \hat{I} de la donnée I , tandis que l'autre terme dépend de l'inconnue C :

$$E_{\text{TV-H}}(C) = \frac{\lambda}{2} \iint \|\Phi_b(\nu_x, \nu_y) [\hat{C}(\nu_x, \nu_y) - \hat{I}(\nu_x, \nu_y)]\|^2 d\nu_x d\nu_y + \iint \|\nabla C(x, y)\|_\epsilon dx dy \quad (3)$$

Le terme d'*attache aux données* de ce nouveau modèle semble effectivement mieux adapté au problème que celui de $E_{\text{ROF}}(C)$, car seules les basses fréquences des images I et C sont contraintes à être égales. Un calcul un peu plus compliqué que celui de l'exercice 3 fournit l'équation d'Euler-Lagrange suivante :

$$\lambda \text{TF}^{-1} \left\{ \Phi_b(\nu_x, \nu_y) [\hat{C}(\nu_x, \nu_y) - \hat{I}(\nu_x, \nu_y)] \right\} (x, y) - \text{Div} \left(\frac{\nabla C(x, y)}{\|\nabla C(x, y)\|_\epsilon} \right) = 0 \quad (4)$$

Écrivez un script, de nom `exercice_4.m`, permettant de résoudre cette équation par un schéma de descente de gradient inspiré du TP7 (contours actifs). Faites ensuite une copie de ce script, de nom `exercice_4_bis.m`, que vous appliquerez à l'image `empreinte.png`.