

TP8 – Décomposition d’une image

La décomposition d’une image u , qui consiste à rechercher deux images \bar{u} et \bar{u}^c de même taille que u , telles que $u = \bar{u} + \bar{u}^c$, est un *problème mal posé*. Il convient donc d’utiliser des connaissances a priori pour garantir l’unicité de la solution, sachant que tout a priori doit être lié au but recherché.

Décomposition structure + texture

La *décomposition structure + texture* d’une image est un des exemples les plus classiques de décomposition d’une image. Les notions de *structure* et de *texture*, qui sont vagues, peuvent être appréhendées sur l’exemple de la figure 1. En pratique, la structure \bar{u} correspond plutôt aux basses fréquences de l’image u , tandis que la texture \bar{u}^c correspond plutôt aux hautes fréquences.

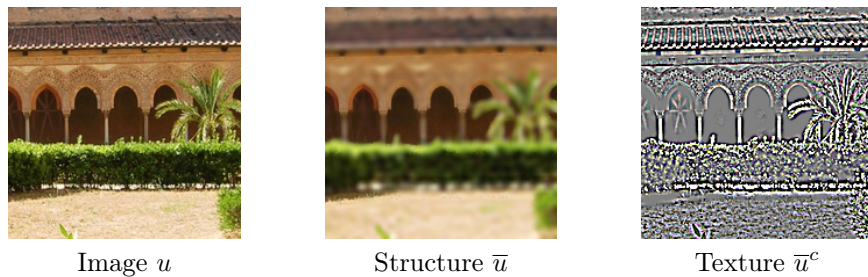


FIGURE 1 – Exemple de décomposition structure + texture provenant du site de tests en ligne IPOL.

Transformation de Fourier discrète (TFD)

L’intérêt de la TFD, vis-à-vis de la décomposition d’une image, vient de ce que la valeur de l’image en un pixel particulier influe sur la totalité du spectre. Réciproquement, la valeur du spectre en un « pixel » particulier influe sur la totalité de l’image. La décomposition d’une image peut donc être menée en trois étapes, en effectuant une partition de son spectre : calcul du spectre s de l’image u ; modification du spectre s , ce qui donne un nouveau spectre \bar{s} ; calcul de la TFD inverse de \bar{s} , ce qui donne une nouvelle image \bar{u} .

Le script `exercice_0` vous permet de tester les trois exemples de modification du spectre d’une grille illustrés sur la figure 2. Ces exemples de décomposition d’une image méritent quelques explications :

- En Matlab, il existe une fonction `fft` pour les signaux 1D (par exemple un son) et une fonction `fft2` pour les signaux 2D (par exemple une image en niveaux de gris).
- La fonction `fftshift` effectue un échange des quatre « quartiers » du spectre. Cela permet au repère Oxy , dont les axes x et y correspondent aux fréquences f_x et f_y , d’avoir son origine O au centre du spectre (par défaut, l’origine se trouve aux quatre coins du spectre !). Inversement, la fonction inverse `ifftshift` doit être appliquée avant le calcul de la TFD inverse `ifft2`.
- L’image comporte des fréquences f_x élevées (rayures verticales) et des fréquences f_y élevées (rayures horizontales). Dans l’exemple 1, le fait de mettre à 0 les fréquences f_x élevées a pour effet de faire disparaître les rayures verticales. Dans l’exemple 2, c’est l’inverse : la mise à 0 les fréquences f_y élevées a pour effet de faire disparaître les rayures horizontales. La direction des rayures conservées est donc orthogonale à la direction de la partie conservée du spectre. Cela explique la direction des rayures de l’exemple 3 (notez quand même que ces rayures ne sont pas aussi nettes que celles des autres images).

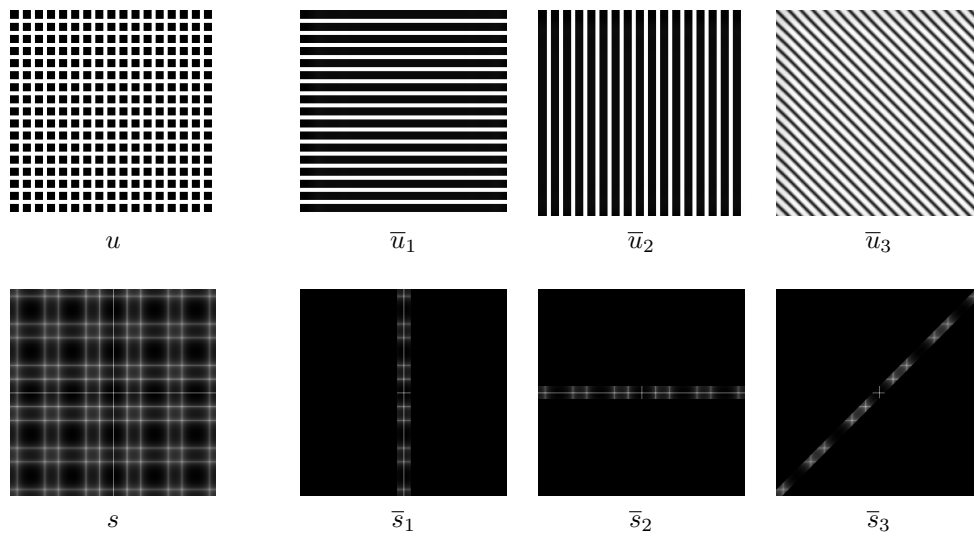


FIGURE 2 – Colonne de gauche : image u d’une grille et son spectre s . Colonnes suivantes : trois exemples de spectres modifiés \bar{s} , où une partie de s a été mise à 0, et les images correspondantes \bar{u} calculées par TFD inverse. Par ce simple procédé, on peut extraire de l’image les rayures horizontales ou les rayures verticales, voire même créer des rayures diagonales qui n’existaient pas dans l’image originale !

- Pour des questions de visibilité, ce n’est pas le spectre $\mathbf{s} = \text{fftshift}(\text{fft2}(u))$ qui est affiché, mais le logarithme de son module complexe $\log(\text{abs}(\mathbf{s}))$ (idem pour les trois autres spectres).
- Pour chacun de ces exemples, si \bar{s}^c désigne le spectre complémentaire de \bar{s} , tel que $\bar{s} + \bar{s}^c = s$, grâce à la linéarité de la TFD inverse, nous en déduisons que $\text{TFD}^{-1}(\bar{s}) + \text{TFD}^{-1}(\bar{s}^c) = \text{TFD}^{-1}(s)$. Autrement dit : $\bar{u} + \text{TFD}^{-1}(\bar{s}^c) = u$, ce qui signifie que \bar{s}^c est le spectre de \bar{u}^c .
- Enfin, s’il est certain que $s \in \text{TFD}(\mathbb{R}^{m \times n})$, de par sa définition, il n’en va généralement pas de même pour le spectre modifié \bar{s} . La TFD inverse de \bar{s} peut comporter une partie imaginaire, qu’il est nécessaire d’éliminer à l’aide de la fonction `real` de Matlab.

Exercice 1 : décomposition d’une image par modification de son spectre

Faites une copie du script `exercice_0`, de nom `exercice_1`, que vous modifierez de manière à effectuer la décomposition structure + texture de l’image `Barbara.png` par modification de son spectre. Pour le spectre de la structure, seuls les pixels situés à une distance inférieure à η (paramètre à ajuster autour de 0,05) doivent être conservés. Veillez à afficher les trois images de la décomposition et les trois spectres qui leur correspondent.

Indication – La création des matrices `f_x` et `f_y` peut être effectuée avec la fonction `meshgrid` de Matlab :

```
[f_x,f_y] = meshgrid(1:nb_colonnes,1:nb_lignes);
f_x = f_x/nb_colonnes-0.5;
f_y = f_y/nb_lignes-0.5;
```

les deux dernières lignes de cette séquence servant à recaler les fréquences dans l’intervalle $[-0,5; 0,5]$.

Le résultat de cette décomposition est plutôt décevant. Une amélioration possible consiste à ne pas effectuer une *partition franche* entre les pixels du spectre, mais à effectuer une *partition douce*, c’est-à-dire à affecter à chaque pixel du spectre de la structure un poids d’autant plus élevé que sa distance à l’origine est plus petite, ce qui tend bien à favoriser les faibles fréquences. Faites une copie du script `exercice_1`, de nom `exercice_1_bis`, dans lequel les pixels du spectre seront pondérés par les valeurs suivantes, qui sont comprises entre 0 et 1 :

$$\Phi(f_x, f_y) = \frac{1}{1 + (f_x^2 + f_y^2)/\eta} \quad (1)$$

Observez l’amélioration des résultats pour différentes valeurs du paramètre η autour de 0,05.

Exercice 2 : décomposition d'une image par le modèle ROF

La décomposition structure + texture d'une image peut également être menée par l'approche variationnelle, sans nécessairement recourir à la TFD. Revenons momentanément à des images continues. La décomposition d'une image u en structure \bar{u} , qui correspond aux basses fréquences, et en texture \bar{u}^c , qui correspond aux hautes fréquences, doit vérifier $u = \bar{u} + \bar{u}^c$. Nous attendons de \bar{u} qu'elle soit « proche » de l'image originale u et que les variations de niveau de gris soient moins brutales dans \bar{u} que dans u . Ces observations nous amènent à utiliser le *modèle ROF* (proposé par Rudin, Osher et Fatemi en 1992), qui fait apparaître la *variation totale* :

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

où $\lambda > 0$ est un paramètre du modèle et $|\nabla \bar{u}(x, y)|_{\epsilon} = \sqrt{|\nabla \bar{u}(x, y)|^2 + \epsilon}$. Ce modèle a déjà été rencontré dans l'exercice 1 du TP5 (modèle de débruitage par variation totale). L'équation d'Euler-Lagrange associée à la fonctionnelle $E_{\text{ROF}}(\bar{u})$ est non linéaire. Elle peut être résolue par un schéma itératif de type « point fixe », qui consiste à calculer la solution à l'itération $k + 1$ en « figeant » la partie non linéaire à l'itération k :

$$\left[I - \lambda \nabla \cdot \left(\frac{1}{\sqrt{|\nabla \bar{u}^{(k)}(x, y)|^2 + \epsilon}} \nabla \right) \right] \bar{u}^{(k+1)}(x, y) = u(x, y) \quad (3)$$

où $\nabla \cdot$ désigne l'opérateur divergence. Après discrétisation, l'itération (3) s'écrit :

$$\left[\mathbf{I}_N - \lambda \left(-\mathbf{D}_x^T \mathbf{W}^{(k)} \mathbf{D}_x - \mathbf{D}_y^T \mathbf{W}^{(k)} \mathbf{D}_y \right) \right] \bar{\mathbf{u}}^{(k+1)} = \mathbf{u} \quad (4)$$

Dans (4), \mathbf{I}_N désigne la matrice identité (N étant le nombre de pixels de Ω) et $\mathbf{W}^{(k)}$ la matrice diagonale des coefficients $\frac{1}{\sqrt{|\nabla \bar{u}^{(k)}(x, y)|^2 + \epsilon}}$ calculés aux différents pixels de Ω . Ces deux matrices de taille $N \times N$ peuvent être calculées à l'aide des fonctions `speye` et `spdiags`, respectivement. Les vecteurs $\bar{\mathbf{u}}^{(k+1)}$ et \mathbf{u} s'obtiennent par concaténation des valeurs de $\bar{u}^{(k+1)}(x, y)$ et $u(x, y)$ aux différents pixels de Ω . Quant aux matrices \mathbf{D}_x et \mathbf{D}_y , elles permettent de calculer les dérivées partielles d'une fonction $f : \Omega \rightarrow \mathbb{R}$ par différences finies, pour peu que le domaine Ω soit rectangulaire. Ces deux matrices bi-diagonales, de taille $N \times N$, se calculent elles aussi à l'aide de la fonction `spdiags` (cf. TP5, TP6 et TP7).

La résolution de l'équation (4), que l'on réécrit $\mathbf{A}^{(k)} \bar{\mathbf{u}}^{(k+1)} = \mathbf{u}$, peut être effectuée très simplement à l'aide de l'opérateur \ de Matlab. Enfin, il est naturel de choisir comme initialisation de cette itération $\bar{\mathbf{u}}^{(0)} = \mathbf{u}$.

Écrivez les fonctions `calcul_gradient` et `calcul_structure_2`, appelées par le script `exercice_2`, qui permettent d'effectuer la décomposition structure + texture d'une image en couleur grâce à 20 itérations du schéma numérique (4), avec $\epsilon = 0,01$ et $\lambda = 100$. Bien entendu, vous pouvez tester ce script sur d'autres images !

Exercice 3 : modèle TV-Hilbert (exercice facultatif)

Il est possible de résoudre le problème de la décomposition structure + texture d'une image en utilisant le *modèle TV-Hilbert*, qui est mixte : le terme d'attache aux données dépend des spectres de u et de \bar{u} , tandis que le terme de régularisation dépend de l'inconnue \bar{u} (TF désigne la transformation de Fourier continue) :

$$E_{\text{TV-H}}(\bar{u}) = \frac{1}{2} \iint_{\hat{\Omega}} |\Phi(f_x, f_y) [\text{TF}(\bar{u})(f_x, f_y) - \text{TF}(u)(f_x, f_y)]|^2 df_x df_y + \mu \iint_{\Omega} |\nabla \bar{u}(x, y)|_{\epsilon} dx dy \quad (5)$$

où $|\cdot|$ désigne le module complexe. L'égalité de Parseval permet de réécrire (5) sous la forme suivante :

$$E_{\text{TV-H}}(\bar{u}) = \iint_{\Omega} \left\{ \frac{1}{2} |\text{TF}^{-1} \{ \Phi [\text{TF}(\bar{u}) - \text{TF}(u)] \}(x, y)|^2 + \mu |\nabla \bar{u}(x, y)|_{\epsilon} \right\} dx dy \quad (6)$$

Si $\Phi \equiv 1$, alors $E_{\text{TV-H}}(\bar{u})$ est identique à la fonctionnelle $E_{\text{ROF}}(\bar{u})$ de l'exercice 2. Ce nouveau modèle semble donc être une version mieux adaptée au problème que $E_{\text{ROF}}(c)$, puisque dans (5), les spectres de u et de \bar{u} sont contraints par le filtre passe-bas $\Phi(f_x, f_y)$ à être égaux uniquement dans les basses fréquences.

Une propriété de la TF et de son inverse est de transformer la multiplication en *convolution* :

$$\begin{aligned} \text{TF}^{-1} \{ \Phi [\text{TF}(\bar{u}) - \text{TF}(u)] \} &= \text{TF}^{-1} \{ \Phi \} * \text{TF}^{-1} \{ [\text{TF}(\bar{u}) - \text{TF}(u)] \} \\ &= \text{TF}^{-1} \{ \Phi \} * [\bar{u} - u] \end{aligned} \quad (7)$$

Comme la fonction Φ définie en (1) ne dépend que de la distance $\sqrt{f_x^2 + f_y^2}$ à l'origine, il s'avère que $\text{TF}^{-1} \{ \Phi \}$ est réelle. Les images \bar{u} et u étant également réelles, nous pouvons donc réécrire (6) sans module complexe :

$$E_{\text{TV-H}}(\bar{u}) = \iint_{\Omega} \left\{ \frac{1}{2} (\text{TF}^{-1} \{ \Phi \} (x, y) * [\bar{u}(x, y) - u(x, y)])^2 + \mu |\nabla \bar{u}(x, y)|_{\epsilon} \right\} dx dy \quad (8)$$

L'équation d'Euler-Lagrange découlant de cette dernière expression de $E_{\text{TV-H}}(\bar{u})$ s'écrit, $\forall (x, y) \in \Omega$:

$$(\text{TF}^{-1} \{ \Phi \} (x, y) * [\bar{u}(x, y) - u(x, y)]) \frac{\partial (\text{TF}^{-1} \{ \Phi \} (x, y) * [\bar{u}(x, y) - u(x, y)])}{\partial \bar{u}} - \mu \nabla \cdot \left(\frac{\nabla \bar{u}(x, y)}{|\nabla \bar{u}(x, y)|_{\epsilon}} \right) = 0 \quad (9)$$

Or, une des nombreuses propriétés de la TF permet d'écrire :

$$\begin{aligned} \frac{\partial (\text{TF}^{-1} \{ \Phi \} (x, y) * [\bar{u}(x, y) - u(x, y)])}{\partial \bar{u}} &= \text{TF}^{-1} \{ \Phi \} (x, y) * \frac{\partial (\bar{u}(x, y) - u(x, y))}{\partial \bar{u}} \\ &= \text{TF}^{-1} \{ \Phi \} (x, y) * \mathbf{1}(x, y) \end{aligned} \quad (10)$$

La convolution de toute fonction par la fonction $\mathbf{1}$ uniformément égale à 1 étant une constante K , nous tirons finalement de l'équation d'Euler-Lagrange (9), en posant $\mu' = \mu/K$:

$$\text{TF}^{-1} \{ \Phi \} (x, y) * [\bar{u}(x, y) - u(x, y)] - \mu' \nabla \cdot \left(\frac{\nabla \bar{u}(x, y)}{|\nabla \bar{u}(x, y)|_{\epsilon}} \right) = 0 \quad (11)$$

En utilisant à nouveau l'égalité (7), (11) s'écrit enfin :

$$\text{TF}^{-1} \{ \Phi [\text{TF}(\bar{u}) - \text{TF}(u)] \} (x, y) - \mu' \nabla \cdot \left(\frac{\nabla \bar{u}(x, y)}{|\nabla \bar{u}(x, y)|_{\epsilon}} \right) = 0 \quad (12)$$

Pour résoudre cette équation, nous utilisons un schéma de descente de gradient, qui consiste à appliquer l'itération suivante (le pas de descente $\gamma > 0$ doit être fixé par l'utilisateur) :

$$\bar{u}^{(k+1)}(x, y) = \bar{u}^{(k)}(x, y) - \gamma \left[\text{TF}^{-1} \{ \Phi [\text{TF}(\bar{u}^{(k)}) - \text{TF}(u)] \} (x, y) - \mu' \nabla \cdot \left(\frac{\nabla \bar{u}^{(k)}(x, y)}{|\nabla \bar{u}^{(k)}(x, y)|_{\epsilon}} \right) \right] \quad (13)$$

Bien qu'elle soit réelle en théorie, en pratique, la fonction $\text{TF}^{-1} \{ \Phi [\text{TF}(\bar{u}^{(k)}) - \text{TF}(u)] \}$ retourne des valeurs comportant une petite partie imaginaire, qu'il convient d'éliminer en ne conservant que la partie réelle. D'autre part, un calcul fastidieux, mais sans difficulté, permet d'explicitier l'expression suivante :

$$\nabla \cdot \left(\frac{\nabla \bar{u}}{|\nabla \bar{u}|_{\epsilon}} \right) = \frac{\bar{u}_{xx}(\bar{u}_y^2 + \epsilon) + \bar{u}_{yy}(\bar{u}_x^2 + \epsilon) - 2\bar{u}_x\bar{u}_y\bar{u}_{xy}}{(\bar{u}_x^2 + \bar{u}_y^2 + \epsilon)^{3/2}} \quad (14)$$

où \bar{u}_x , \bar{u}_y , \bar{u}_{xx} , \bar{u}_{xy} et \bar{u}_{yy} désignent les dérivées partielles d'ordres 1 et 2 de la fonction \bar{u} . Étant donné que le domaine Ω est rectangulaire, ces dérivées partielles peuvent être calculées à l'aide des matrices \mathbf{D}_x et \mathbf{D}_y :

$$\bar{\mathbf{u}}_x = \mathbf{D}_x \bar{\mathbf{u}} \quad ; \quad \bar{\mathbf{u}}_y = \mathbf{D}_y \bar{\mathbf{u}} \quad ; \quad \bar{\mathbf{u}}_{xx} = -\mathbf{D}_x^{\top} \mathbf{D}_x \bar{\mathbf{u}} \quad ; \quad \bar{\mathbf{u}}_{xy} = -\mathbf{D}_x^{\top} \mathbf{D}_y \bar{\mathbf{u}} \quad ; \quad \bar{\mathbf{u}}_{yy} = -\mathbf{D}_y^{\top} \mathbf{D}_y \bar{\mathbf{u}}$$

où $\bar{\mathbf{u}}$, $\bar{\mathbf{u}}_x$, $\bar{\mathbf{u}}_y$, $\bar{\mathbf{u}}_{xx}$, $\bar{\mathbf{u}}_{xy}$ et $\bar{\mathbf{u}}_{yy}$ désignent des matrices vectorisées.

Écrivez les fonctions `calcul_Phi` et `calcul_structure_3`, appelées par le script `exercice_3`, permettant d'effectuer la décomposition `structure + texture` d'une image en couleur grâce à 1000 itérations de la version discrète du schéma numérique (13), avec $\epsilon = 0,5$, $\mu' = 5000$ et $\gamma = 3.10^{-5}$. Pour ne pas trop ralentir l'exécution de ce script, il est préférable de n'effectuer la mise à jour des affichages que toutes les 50 itérations.