

Traitements et outils de base

Le traitement d'image (*image processing*) consiste à appliquer des transformations mathématiques sur des images dans le but d'améliorer leur qualité ou d'en extraire une information. On peut citer par exemple : la création d'effets spéciaux au cinéma, l'amélioration de la qualité des photos en corrigeant les conditions de prise de vue, la détection de lettres ou de visages, l'identification de zones cancéreuses en imagerie médicale, la compression JPEG, etc.

Ce document présente succinctement les outils élémentaires du traitement d'images. Il est indispensable de l'étudier pour aborder le premier TP dans de bonnes conditions. Ce premier TP commencera par une rapide évaluation (notée) de vos pré-requis, dont vous trouverez des questions page 13.

Dans ce document, les termes en anglais sont donnés en *italique*.

1	Définition d'une image numérique	2
1.1	Définition	2
1.2	Diversité des images	2
1.3	Palette de couleurs	3
2	Opérations arithmétiques	3
2.1	Addition d'images	3
2.2	Soustraction d'images	4
2.3	Division d'images	4
3	Histogramme	4
3.1	Définition	4
3.2	Propriétés	5
3.3	Transformations d'intensité	5
3.4	Seuillage	7
4	Convolution	7
4.1	Définition	7
4.2	Propriétés	9
4.3	Problèmes aux bords	9
4.4	Séparabilité	10
5	Transformée de Fourier	10
5.1	Définition	10
5.2	Propriétés	11
5.3	Filtrage	11
6	Questions de l'évaluation	13

1 Définition d'une image numérique

1.1 Définition

Une image numérique est définie comme une fonction $f(m, n, \dots)$ à valeurs discrètes et finies (mais pas forcément scalaires), et dont les coordonnées $m, n, \dots$ sont également à valeurs discrètes et finies.

$$\begin{aligned} f : \quad \mathbb{N}^d &\rightarrow \mathbb{R}^B \\ m, n, \dots &\mapsto f(m, n, \dots) \end{aligned}$$

Ainsi par exemple, une image à niveaux de gris correspond à $d = 2$ (l'image a deux dimensions) et $B = 1$ (il n'y a qu'une valeur par pixel : le niveau de gris). Pour une image couleur classique, $d = 2$ et $B = 3$ (il y trois bandes : rouge, vert, bleu). Enfin, une image IRM correspond à $d = 3$ (l'image est 3D) et $B = 1$.

Dans le cas classique d'une image à deux dimensions $f(m, n)$ de taille $M \times N$, on adopte le système de coordonnées représenté figure 1 : le pixel $(0, 0)$ est situé en haut à gauche.

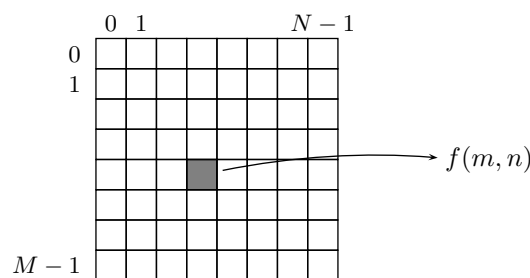


FIGURE 1 – Système de coordonnées adopté dans le cours.

1.2 Diversité des images

Les images numériques peuvent se distinguer sur plusieurs aspects.

Nombre de dimensions Une image classique, telle une photographie, possède deux dimensions, mais ce n'est pas le cas de toutes les images. Par exemple, une image IRM est une image à trois dimensions (qu'on appelle alors « image 3D » ou « cube »), et une image à une dimension est en fait un signal. Les éléments d'une image 2D sont appelés pixels (*picture element*) et ceux d'une image 3D sont appelés voxels (*volume element*).

Hétérogénéité des dimensions Dans une image classique 2D, les deux dimensions de l'image sont des dimensions spatiales. Mais là encore, les différentes dimensions de l'image ne s'expriment pas forcément dans la même unité. Par exemple, une vidéo peut être considérée comme une image 2D+t (deux dimensions spatiales, une dimension temporelle), une séquence IRM peut être considérée comme une image 3D+t (trois dimensions spatiales, une dimension temporelle) et une image hyperspectrale est une image 2D+λ (deux dimensions spatiales plus une dimension dépendant de la longueur d'onde).

Dimension de leurs éléments Chaque élément d'une image (pixel, voxel...) peut être scalaire ou vectoriel. Par exemple, les pixels d'une image 2D à niveaux de gris (*grayscale image*) ne contiennent qu'une seule valeur (l'intensité du gris). Les pixels d'une photographie classique contiennent trois valeurs (correspondant aux proportions de rouge, de vert et de bleu). Les images issues des satellites Pléiades sont des images RVB-IR, donc contiennent quatre valeurs (rouge, vert, bleu et infrarouge).

Espace des valeurs de leurs éléments les images grand public sont généralement à valeurs dans $\{0, 1, \dots, 255\}$, mais une « image binaire » (*binary image*) est à valeurs dans $\{0, 1\}$. De manière générale, on considère qu'une image est à valeurs dans $\{i_1, \dots, i_L\}$ où $i_l \in \mathbb{R}$ et $L \in \mathbb{N}^*$.

1.3 Palette de couleurs

Pour afficher une image, il faut définir une palette de couleur (*colormap*) qui fait la correspondance entre les intensités $\{i_1, \dots, i_L\}$ des pixels et les niveaux de gris ou les couleurs. La figure 2 représente l'image de Lena¹ avec différentes palettes. L'image de gauche correspond à l'image originale avec la palette par défaut (256 niveaux de 0 à 255). Comme on le voit, le choix de la palette permet de modifier la perception des couleurs.



FIGURE 2 – L'image Lena représentée avec les palettes situées sous les images.

2 Opérations arithmétiques

Des opérations mathématiques simples peuvent être effectuées sur les éléments de deux images. Pour simplifier, on considère dans cette partie des images 2D, la généralisation à des images de dimension différente est évidente.

2.1 Addition d'images

L'addition de deux images f et g de même taille est une nouvelle image h de même taille dont chaque pixel correspond à la somme des valeurs des pixels des images originales :

$$\forall m, n, \quad h(m, n) = f(m, n) + g(m, n).$$

La figure 3 donne un exemple d'utilisation de l'addition. L'addition peut également être utilisée pour débruiter une série d'images, comme on le verra dans le cours « Restauration ».



FIGURE 3 – L'image de droite est la somme des deux images de gauche. Notez que cette somme a des valeurs entre 0 et 510.

1. Pour en savoir plus sur Lena : fr.wikipedia.org/wiki/Lenna ou www.cs.cmu.edu/~chuck/lennapg/.

2.2 Soustraction d'images

La soustraction de deux images est utilisée par exemple pour détecter des changements (figure 4).

$$\forall m,n, \quad h(m,n) = f(m,n) - g(m,n)$$

ou $\forall m,n, \quad h(m,n) = |f(m,n) - g(m,n)|$



FIGURE 4 – L'image de droite est la différence des deux images de gauche. Notez que l'image de différence a des valeurs entre -255 et 255 .

2.3 Division d'images

La division de deux images permet notamment de corriger une illumination non homogène [3, section 10.3.2]. La figure 5 illustre la suppression de l'ombre sur une image.

$$\forall m,n, \quad h(x,y) = \frac{f(m,n)}{g(m,n)}$$

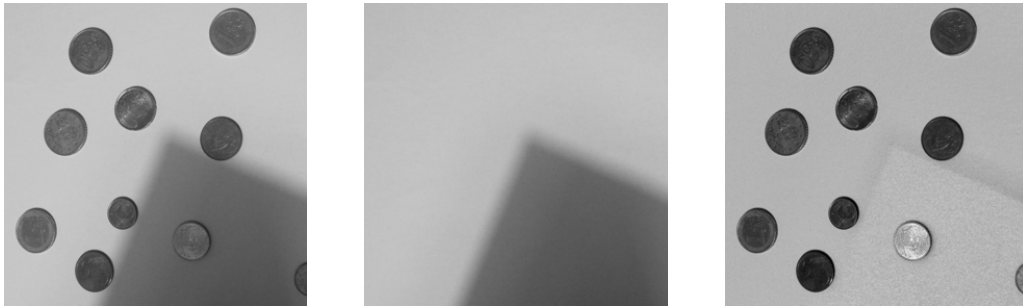


FIGURE 5 – L'image de droite est la division de l'image de gauche par l'image de droite.

3 Histogramme

3.1 Définition

L'histogramme (*histogram*) d'une image numérique représente la distribution des valeurs des pixels. C'est la fonction discrète h telle que

$$h(i) = n_i$$

où n_i est le nombre de pixels de l'image ayant l'intensité i .

La figure 6 représente une image et son histogramme. On distingue sur celui-ci deux « modes » : celui de gauche correspond aux tons sombres de l'image (principalement le fond) et celui de droite correspond aux tons clairs (les pétales et le centre).

Représentez l'histogramme de la figure 7. Les chiffres correspondent aux intensités des pixels.

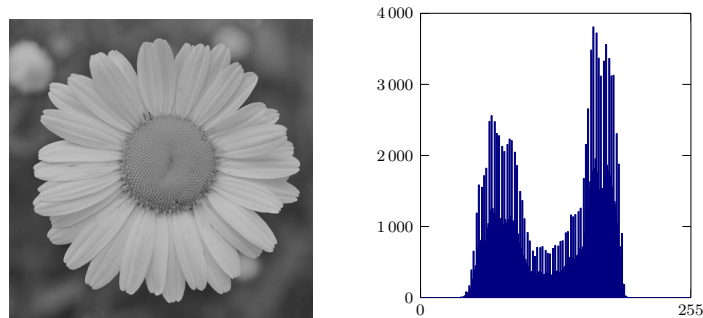


FIGURE 6 – Une image à 256 niveaux de gris et son histogramme.

3	2	2	1
0	3	0	3
3	2	0	0
0	1	1	1

FIGURE 7 – Une simple image : représentez son histogramme !

3.2 Propriétés

- L'histogramme peut être interprété comme la densité de probabilité discrète des intensités si les effectifs sont normalisés par le nombre de pixels $M \times N$:

$$p(i) = \frac{n_i}{M \times N}$$

- L'histogramme donne une information globale sur les intensités de l'image, mais perd l'information spatiale de l'image. Ainsi, deux images très différentes peuvent avoir le même histogramme (cf. figure 8).



FIGURE 8 – Ces deux images ont le même histogramme (représenté figure 6). L'image de droite correspond en fait aux pixels de l'image de gauche ordonnés selon leur niveau de gris.

- Le nombre et la largeur des barres (*bins*) est choisi par l'utilisateur.

3.3 Transformations d'intensité

On peut appliquer une transformation T sur les intensités des pixels d'une image pour en modifier les valeurs [3, section 10.2.3] :

$$j = T(i)$$

où j et i représentent respectivement les intensités des pixels de la nouvelle image et de l'image originale. L'application d'une telle transformation a alors une incidence sur l'histogramme (cf. figure 9).

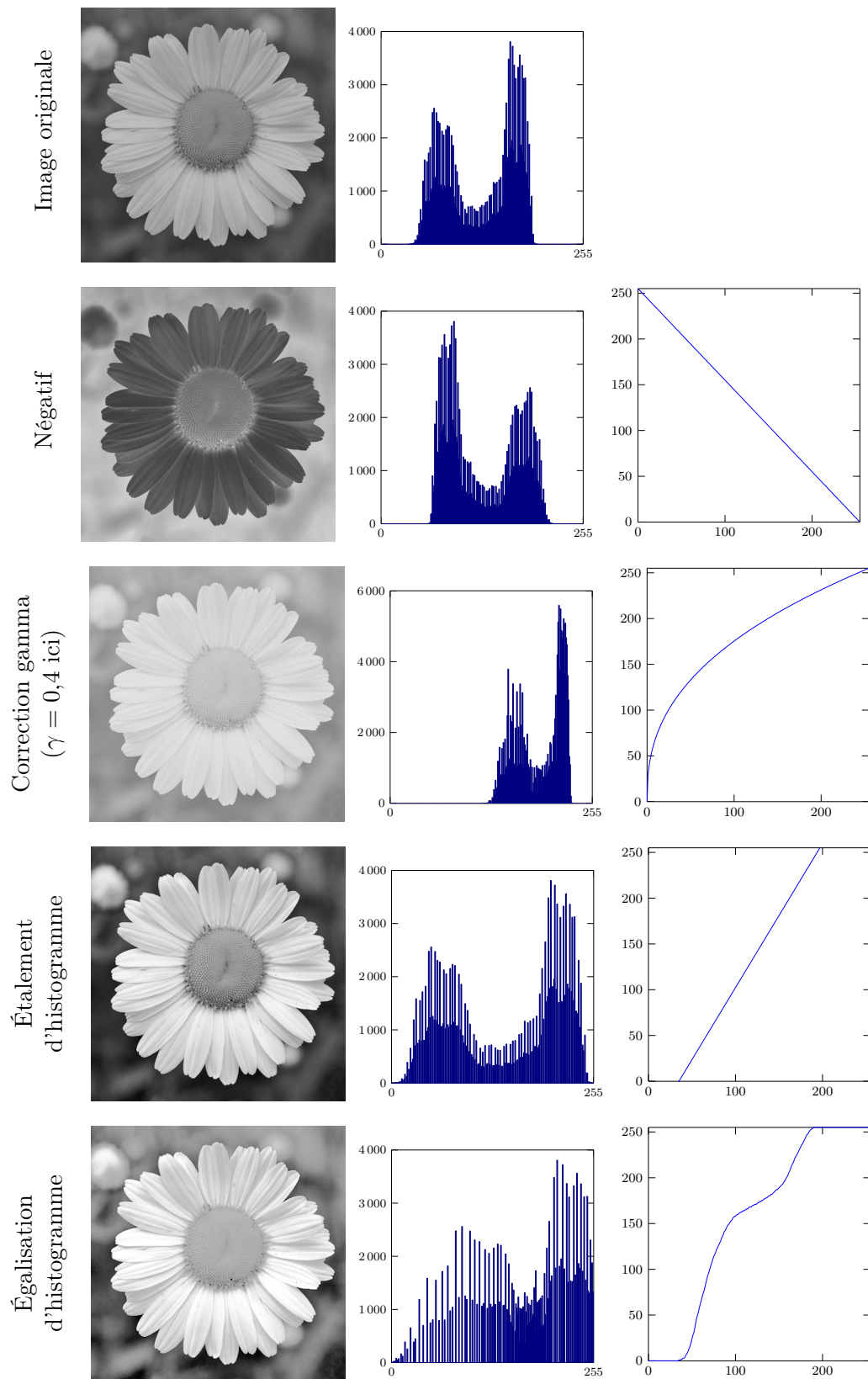


FIGURE 9 – Exemples de transformations. De gauche à droite : image (originale ou transformée) et son histogramme, transformation T .

Quatre transformations classiques sont répertoriées ci-après (on suppose que les intensités sont à valeurs dans $[0,1]$).

- négatif : $T(i) = 1 - i$;
- correction gamma : $T(i) = i^\gamma$;
- étalement (*normalization*) :

$$T(i) = \frac{i - i_{\min}}{i_{\max} - i_{\min}}$$

où $i_{\min}$ et $i_{\max}$ sont respectivement les intensités minimale et maximale de l'image originale ;

- égalisation (*equalization*) :

$$T(i) = \frac{1}{MN} \sum_{k=0}^i n_k$$

où M et N sont les dimensions de l'image et n_k est le nombre de pixels d'intensité k . Cette transformation cherche à étaler au mieux l'histogramme sur toute la dynamique des intensités, et à rendre l'histogramme le plus plat possible : cela a pour conséquence d'augmenter le contraste de l'image. C'est une méthode complètement automatique qui ne nécessite aucun paramètre à régler. La démonstration de cette équation est disponible dans [1, section 3.3.1].

3.4 Seuillage

Dans certains cas, l'histogramme peut être un outil efficace pour segmenter une image en deux classes, c'est-à-dire pour distinguer les objets de l'image suivant leur luminosité. En effet, lorsque l'histogramme présente deux modes distincts, on peut définir un seuil (*threshold*) S entre ces deux modes, et appliquer ensuite un seuillage (*thresholding*) sur les pixels de l'image :

- si le pixel a une valeur inférieure au seuil, le pixel est dans la classe 0 ;
- si le pixel a une valeur supérieure au seuil, le pixel est dans la classe 1.

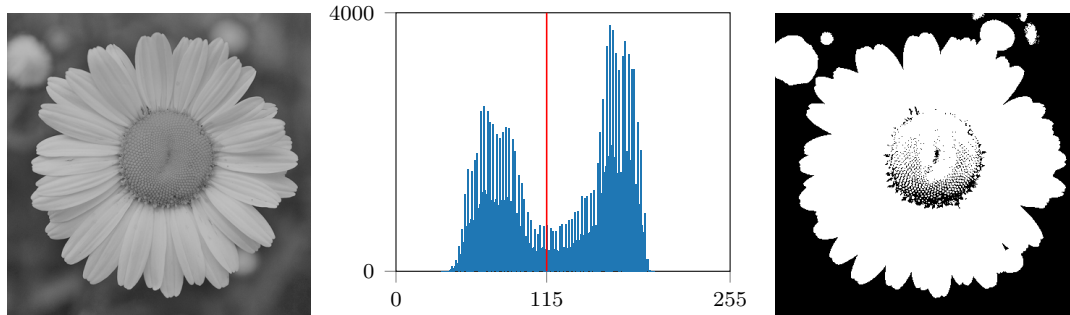


FIGURE 10 – Seuillage avec un seuil égal à 115.

Le seuillage permet d'obtenir une image binaire qui ne contient que deux valeurs. Il existe des méthodes pour calculer automatiquement le seuil (par exemple avec la méthode de Otsu comme on le verra dans le cours « Segmentation »).

4 Convolution

4.1 Définition

Beaucoup de traitements s'obtiennent en modifiant les valeurs des pixels en fonction de leurs pixels voisins. Lorsque cette modification est identique à toute une image g , elle peut être définie à l'aide d'une seconde image h qui définit les relations de voisinage. Il en résulte donc une troisième image f . Cette opération est appelée convolution (*convolution*) [2], [3, section 4] et se note $*$:

$$f(x,y) = (g * h)(x,y) = \sum_m \sum_n g(x-m, y-n) h(m,n)$$

$f_{1,1}$	$f_{1,2}$	$f_{1,3}$	$f_{1,4}$	$f_{1,5}$
$f_{2,1}$	$f_{2,2}$	$f_{2,3}$	$f_{2,4}$	$f_{2,5}$
$f_{3,1}$	$f_{3,2}$	$f_{3,3}$	$f_{3,4}$	$f_{3,5}$
$f_{4,1}$	$f_{4,2}$	$f_{4,3}$	$f_{4,4}$	$f_{4,5}$
$f_{5,1}$	$f_{5,2}$	$f_{5,3}$	$f_{5,4}$	$f_{5,5}$

 $=$

$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$
$g_{2,1}$	$g_{2,2}$	$g_{2,3}$	$g_{2,4}$	$g_{2,5}$
$g_{3,1}$	$g_{3,2}$	$g_{3,3}$	$g_{3,4}$	$g_{3,5}$
$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$	$g_{4,5}$
$g_{5,1}$	$g_{5,2}$	$g_{5,3}$	$g_{5,4}$	$g_{5,5}$

 $*$

-1	0	$+1$
$h_{-,-}$	$h_{-,0}$	$h_{-,+}$
$h_{0,-}$	$h_{0,0}$	$h_{0,+}$
$h_{+,-}$	$h_{+,0}$	$h_{+,+}$

 $\begin{matrix} -1 \\ 0 \\ +1 \end{matrix}$

$$f_{2,2} = g_{3,3}h_{-,-} + g_{3,2}h_{-,0} + g_{3,1}h_{-,+} + g_{2,3}h_{0,-} + g_{2,2}h_{0,0} + g_{2,1}h_{0,+} + g_{1,3}h_{+,-} + g_{1,2}h_{+,0} + g_{1,1}h_{+,+}$$

FIGURE 11 – Exemple de calcul du pixel (2,2) de f .

Intuitivement, le résultat d'une convolution peut s'obtenir en faisant s'« étaler » chaque pixel (m,n) de l'image g suivant h et proportionnellement à l'intensité de $g_{m,n}$. La figure 11 donne un exemple de calcul pour un pixel particulier de f . La figure 12 donne quelques exemples de résultats de convolution.

Pour des raisons de simplicité, l'image h est :

- de taille impaire (3×3 , 5×5 , 7×7 , ...);
- centrée, c'est-à-dire que le pixel de coordonnées $(0,0)$ est situé au milieu de l'image.

Notons que h est une image qui porte plusieurs noms, suivant le contexte : filtre (*filter*), masque (*mask*), noyau (*kernel*), fenêtre (*window*), motif (*pattern*) ou fonction d'étalement (*point spread function* : *PSF*).

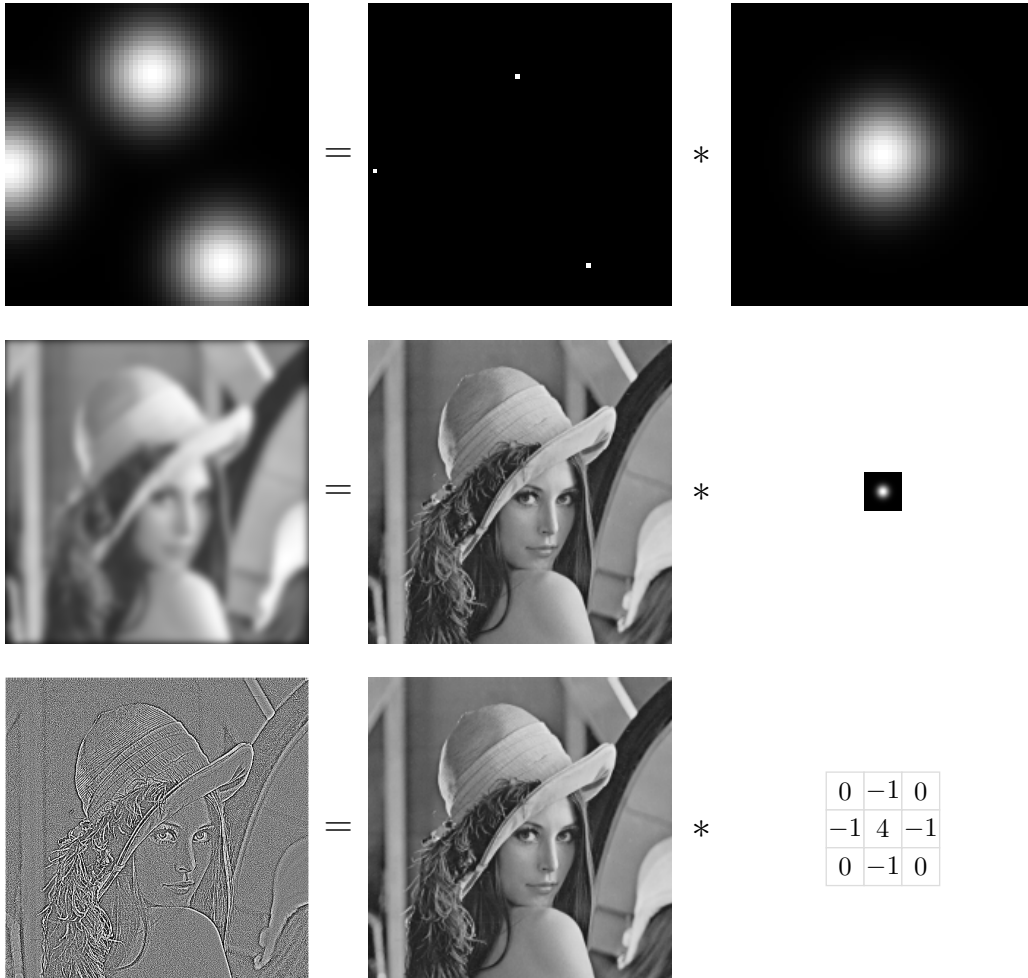


FIGURE 12 – Quelques exemples de produits de convolution.

4.2 Propriétés

- L'élément neutre du produit de convolution est une image nulle avec un seul pixel égal à 1.
- Le produit de convolution est commutatif : $g * h = h * g$.
- Le produit de convolution est distributif par rapport à l'addition : $g * (h_1 + h_2) = g * h_1 + g * h_2$.
- Le produit de convolution est bilinéaire : $\alpha(g * h) = (\alpha g) * h = g * (\alpha h) \quad (\alpha \in \mathbb{C})$.
- Le produit de convolution est associatif : $h_1 * (h_2 * h_3) = (h_1 * h_2) * h_3$.

4.3 Problèmes aux bords

La formule du produit de convolution n'est pas définie sur les bords de l'image : ainsi, le calcul de $f_{1,1}$ dans la figure 11 nécessite de connaître, par exemple, la valeur de $g_{0,0}$ qui n'existe pas. Il existe donc plusieurs manières de fixer les valeurs des pixels situés en dehors de l'image : la figure 13 représente l'image Lena et les différentes possibilités de définir les pixels extérieurs, et la figure 14 présente les résultats de convolution dans les différents cas.

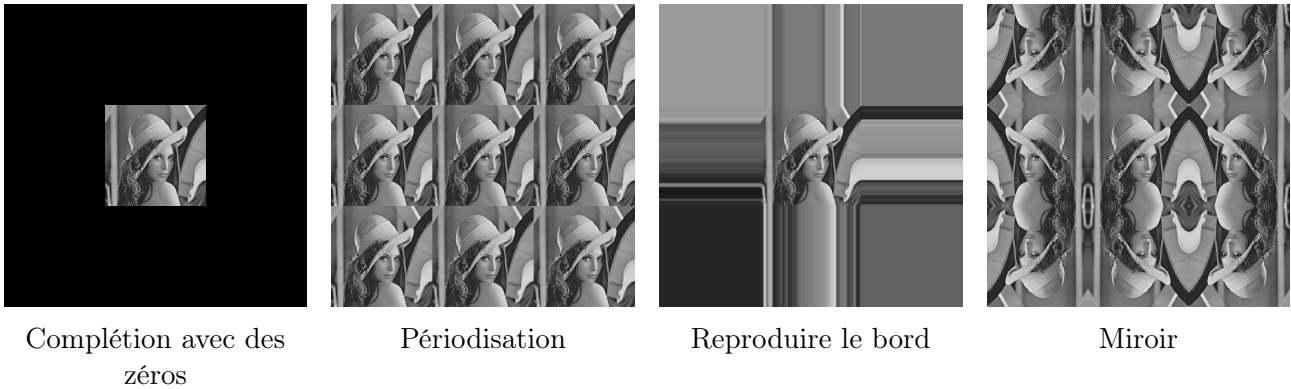


FIGURE 13 – Plusieurs manières de fixer les pixels situés en dehors de l'image Lena.

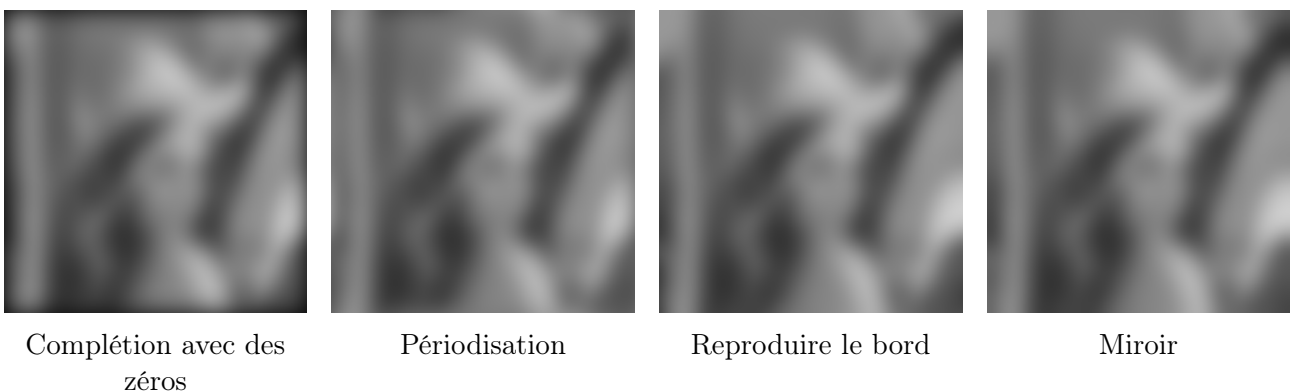


FIGURE 14 – Résultats des convolutions dans les différents cas.

Cette dernière figure montre que, globalement, les résultats sont très proches : seuls les pixels aux bords de l'image peuvent être différents. En tous les cas, il n'existe pas de manière parfaite pour fixer les valeurs des pixels situés en dehors de l'image : toutes introduisent des erreurs. Aussi, le mieux est de s'arranger pour que les objets d'intérêt soient loin du bord. Notons enfin que la périodisation de l'image aboutit à une *convolution circulaire* ; c'est également le résultat obtenu par une multiplication dans le domaine de Fourier (cf. section 5).

4.4 Séparabilité

Lorsqu'un filtre h peut s'écrire comme la convolution de deux filtres 1D (h_1 et h_2) suivant les deux axes, il est dit *séparable*. Un exemple est représenté ci-dessous :

$$\underbrace{\begin{bmatrix} \alpha a & \alpha b & \alpha c \\ \beta a & \beta b & \beta c \\ \gamma a & \gamma b & \gamma c \end{bmatrix}}_h = \underbrace{\begin{bmatrix} 0 & \alpha & 0 \\ 0 & \beta & 0 \\ 0 & \gamma & 0 \end{bmatrix}}_{h_1} * \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ a & b & c \\ 0 & 0 & 0 \end{bmatrix}}_{h_2} = \underbrace{\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}}_{h_1} * \underbrace{\begin{bmatrix} a & b & c \end{bmatrix}}_{h_2}$$

Ainsi, la convolution d'une image g par un filtre séparable h peut être calculée en effectuant tout d'abord la convolution de g par h_1 , puis ensuite par h_2 (ou l'inverse) :

$$g * h = g * (h_1 * h_2) = (g * h_1) * h_2 = (g * h_2) * h_1$$

La séparabilité permet ainsi de gagner en temps de calcul, car le calcul de deux convolutions 1D demande moins d'opérations que le calcul d'une convolution 2D.

5 Transformée de Fourier

La transformée de Fourier (*Fourier transform*) bidimensionnelle est la version étendue de la transformée de Fourier classique aux images [3, section 2.3]. On rappelle que la transformée de Fourier permet de décomposer un signal en une somme de sinusoides, permettant ainsi de mettre en évidence les fréquences contenues dans un signal.

5.1 Définition

La transformée de Fourier discrète (TFD) d'une image f de taille $M \times N$ est une image F de taille $M \times N$, à valeurs complexes :

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(m,n) e^{-j 2\pi (\frac{um}{M} + \frac{vn}{N})}$$

Comme la transformée de Fourier est (généralement) complexe, elle ne peut pas être représentée directement : c'est pourquoi on représente séparément son module et sa phase, comme dans la figure 15.

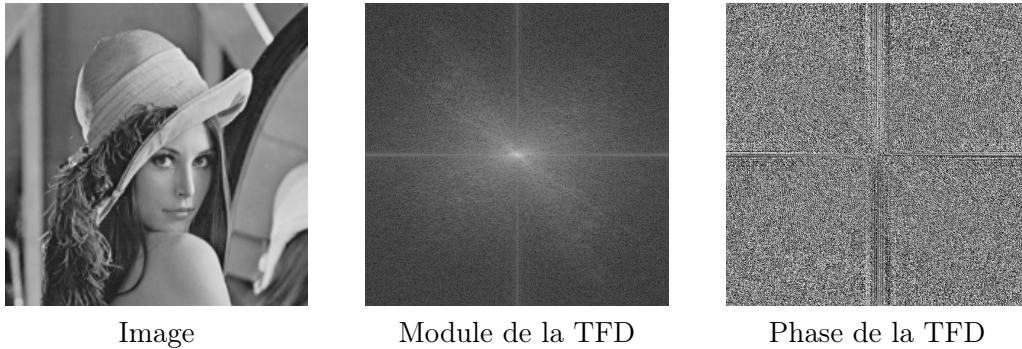


FIGURE 15 – Transformée de Fourier discrète de Lena. Le module est représenté en échelle logarithmique afin de distinguer les détails (on a donc appliqué une transformation d'histogramme!).

Le module et la phase représentent la distribution de l'énergie dans le plan des fréquences. Les fréquences basses se situent au centre de l'image, les fréquences hautes sur le pourtour. Dans l'image « spatiale », l'épaule de Lena est une zone de basses fréquences car les intensités des pixels évoluent doucement d'un pixel à l'autre. Au contraire, les plumes du chapeau sont une zone contenant des hautes fréquences. La transformée de Fourier discrète inverse permet de calculer l'image originale à partir d'une transformée de Fourier :

$$f(m,n) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{+j 2\pi (\frac{um}{M} + \frac{vn}{N})}$$

5.2 Propriétés

- Séparabilité : la TFD peut être calculée en calculant les TFD 1D des lignes, puis les TFD 1D des colonnes résultantes (ou l'inverse).
- La TFD est périodique de périodes M et N :

$$F(u, v) = F(u + k_m M, v) = F(u, v + k_n N) = F(u + k_m M, v + k_n N) \quad \text{où } k_m, k_n \in \mathbb{Z}.$$

- La TFD est linéaire :

$$af + bg \rightarrow aF + bG \quad \text{où } a, b \in \mathbb{C}.$$

- Une translation sur l'image implique un déphasage de sa TFD :

$$f(m - m_0, n - n_0) \rightarrow F(u, v) \exp \left(-j2\pi \left(\frac{um_0}{M} + \frac{vn_0}{N} \right) \right)$$

- Une rotation appliquée sur l'image implique la même rotation sur sa TFD.
- Le produit de convolution de deux images est équivalent à la multiplication de leurs TFD (et vice-versa) :

$$f * g \rightarrow F \times G \quad \text{et} \quad f \times g \rightarrow F * G$$

5.3 Filtrage

Le filtrage correspond à sélectionner certaines fréquences de l'image. On distingue notamment le filtrage passe-bas (*low-pass filtering*) (figure 16) et le filtrage passe-haut (*high-pass filtering*) (figure 17).

Références

- [1] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson, 2010.
- [2] Machine Learning Guru. Image filtering. machinelearningguru.com/computer_vision/basics/convolution/image_convolution_1.html.
- [3] B. Jähne. *Digital Image Processing*. Springer, 2005. Accessible depuis le réseau de l'Université : www.springer.com/gp/book/9783540240358.

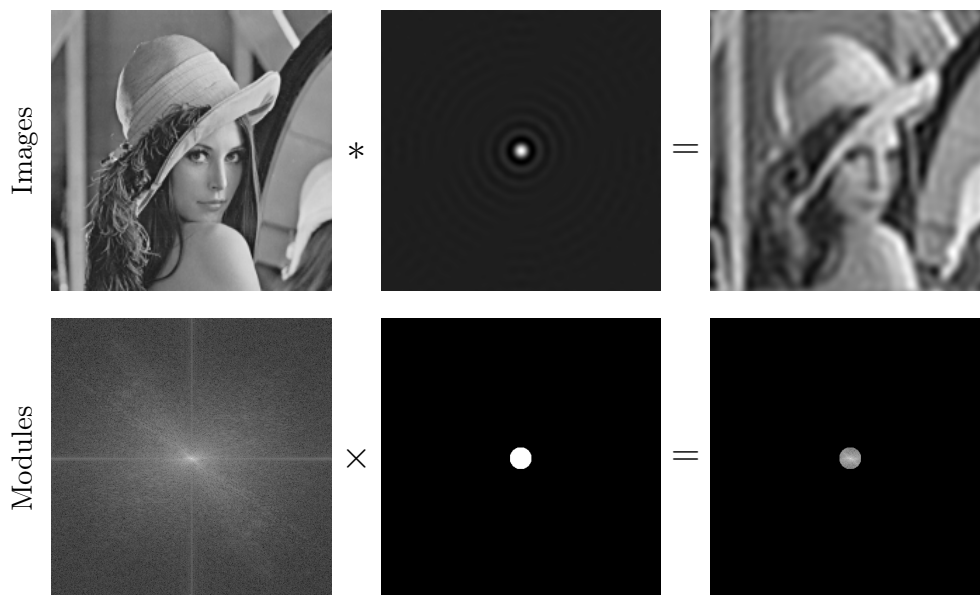


FIGURE 16 – Exemple de filtrage passe-bas de Lena : seules les basses fréquences sont conservées. Les images du haut correspondent aux images dans le domaine spatial, celles du bas correspondent aux modules des TFD des images du haut.

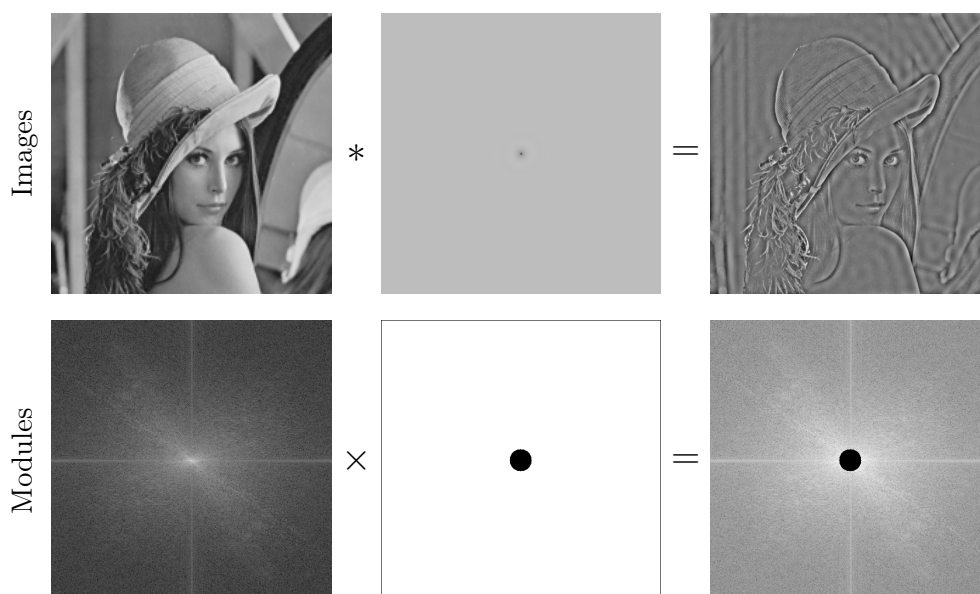


FIGURE 17 – Exemple de filtrage passe-haut de Lena : seules les hautes fréquences sont conservées.

6 Questions de l'évaluation

Les questions de la première évaluation seront tirées de la liste ci-dessous. Cette liste vous permet donc de connaître votre niveau pour aborder sereinement le module d'Outils fondamentaux en traitement d'images. Les réponses ne sont pas forcément toutes dans ce document : à vous de vous renseigner avec d'autres sources. Pour toute question, n'hésitez pas à me contacter : vincent.mazet@unistra.fr.

1. Savoir traduire en français et en anglais les termes utilisés dans ce document.
2. Les écrans utilisent uniquement trois couleurs de pixels (rouge, vert et bleu) pour générer toutes les couleurs par synthèse additive. Comment obtient-on du noir ? du blanc ? du jaune ?
3. Les images classiques sont enregistrées sur trois canaux, chacun codé sur 8 bits. Combien de couleurs sont possibles ? Quelle est la taille en bits d'une image de 1000×1000 pixels ?
4. Comment peut-on représenter une image hyperspectrale sur un écran ?
5. Pourquoi faut-il définir une table de correspondance entre les valeurs des pixels et leur couleur ?
6. Donnez un exemple d'application dans laquelle on souhaite faire une détection de changements entre deux images. Quelle technique peut-on alors utiliser pour détecter les changements ?
7. Qu'obtient-on en divisant une image f par une image g ?
8. Seriez-vous capable d'associer une image et son histogramme parmi différentes possibilités ?
9. Représentez l'histogramme de la figure 7 (ou d'une figure équivalente).
10. Que représentent les deux axes d'un histogramme ?
11. Quelle type de transformation d'histogramme utiliser pour augmenter le contraste d'une image ? Et pour inverser les couleurs d'une image ?
12. Quelle est la différence principale entre un étalement et une égalisation d'histogramme ?
13. Quel est l'effet du seuillage d'une image sur son histogramme ?
14. Comment choisir le seuil pour effectuer une segmentation binaire ?
15. Quelle est la signification physique de la convolution ?
16. Seriez-vous capable d'associer le résultat d'une convolution à deux images d'entrée parmi différentes possibilités ?
17. Comment se traduit une convolution dans le domaine fréquentiel ?
18. Une image f est d'abord convoluée par une image g puis par une image h . Obtient-on le même résultat en effectuant les convolutions dans un ordre différent ?
19. Lorsqu'on a affaire à un problème dans lequel apparaît une convolution, il peut être intéressant de se poser la question de la séparabilité du noyau : pourquoi ?
20. Listez les propriétés du produit de convolution.
21. Pourquoi la transformée de Fourier d'une image est-elle représentée avec deux images ?
22. Pourquoi appliquer une transformation d'intensité sur le module de la transformée de Fourier ?
23. La transformée de Fourier de Lena (de taille 512×512 pixels) est représentée figure 15. À quoi ressemble la transformée de Fourier de Lena redimensionnée à 128×128 pixels ?
24. À quoi correspond l'opération de filtrage dans le domaine spatial ? Et dans le domaine fréquentiel ?
25. Quelle opération permet de flouter une image ?
26. Listez les propriétés de la transformée de Fourier.
27. Quelles sources avez-vous utilisées pour compléter vos connaissances afin de répondre aux questions ci-dessus ?