# Travaux dirigés Traitement d'images n°4

**Filtrage Spatial**

*—Master 1—*

## ▸ Exercice 1. *Filtres de convolution*

1. *À partir du code source en ligne, implémentez une fonction* `cv::mat derivative_x()` *qui renvoie la matrice suivante :*

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

   *Attention, les éléments de cette matrices doivent être du type* `double`*.*

2. *Rappelez pourquoi cette convolution représente une dérivée selon l'axe des $x$.*

3. *Lire et comprendre la fonction* `apply_convolution`*.*
   ***Attention :*** *vous noterez que certaines convolutions, comme la dérivée, peuvent renvoyer des valeurs négatives. Dans ce cas, on ajoutera un offset de 128 au résultat attendu. Pour les filtres plus standard, cet offset sera nul.*

4. *Appliquez la fonction* `apply_convolution` *avec la matrice de convolution de générée par la fonction* `cv::mat derivative_x()` *(avec un offset de 128).*

5. *Est ce que la matrice de convolution suivante calcule aussi une dérivée selon l'axe des $x$ :*

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

   *Pourquoi ?*

6. *Et la matrice de Sobel ?*

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

7. *Faites une fonction* `cv::mat derivative_y()` *et testez la.*

8. *Faites une fonction* `cv::mat laplacian()` *correspondant au calcul de la dérivée seconde de votre image et qui renvoie la matrice suivante :*

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

   *Comparer le résultat avec la dérivée première de la question* **??**.

9. *Un réhausseur de contour consiste à ajouter à une image l'image de ses contours. En pratique, on fait plutôt la soustraction suivante :*

$$\mathrm{I}'(x,y) = \mathrm{I}'(x,y) - \mathrm{Laplacian}\big(\mathrm{I}'(x,y)\big)$$

*Comment vu en cours, la matrice de convolution résultante est :*

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

***Attention :*** *pas d'offset sur cette convolution !*
*Faite un test sur une image, que constatez vous sur les contours?*

### ▸ Exercice 2. *Flou*

1. *Dans la continuité de l'exercice précédent, implémentez une matrice de convolution moyenneur de taille $n \times n$ :* `cv::Mat average(const unsigned int n)`. *Pour rappel, un filtre moyenneur de taille $N \times N$ prend la forme :*

$$h = \frac{1}{N^2}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

2. *Appliquer le filtrage moyenneur sur une images de test avec différents $N$ (impairs) et comparer les résultats obtenus.*

3. *Faites une fonction qui renvoie une matrice $5 \times 5$ proche de ce que vous renverait une gaussienne. Que se passe-t-il si la somme de vos coefficients n'est pas égale à 1 ?*

### ▸ Exercice 3. *Filtre médian*

1. *Implémentez un filtre médian.*

2. *Testez le sur une image bruitée.*

### ▸ Exercice 4. *Image intégrale*

*L'"Image intégrale" est un algorithme permettant de calculer rapidement des sommes de valeurs dans des zones rectangulaires d'une image numérique. Cet algorithme calcule une image (intégrale) $\hat{\mathrm{I}}$, de même taille que l'image d'origine $\mathrm{I}$, qui en chacun de ses points contient la somme des pixels situés au-dessus et à gauche de ce point. Mathématiquement, l'image intégrale $\hat{\mathrm{I}}$ est définie comme :*

$$\hat{\mathrm{I}}(x,y) = \sum_{x' \le x} \sum_{y' \le y} \mathrm{I}(x',y')$$

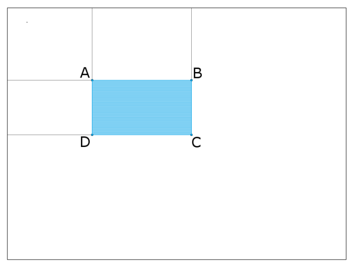
*En pratique, l'image intégrale $\hat{I}$ se calcule en utilisant un seul parcours de l'image d'origine* $\mathrm{I}$ *:*

$$\hat{\mathrm{I}}(x,y) = I(x,y) + \hat{\mathrm{I}}(x-1,y) + \hat{\mathrm{I}}(x,y-1) - \hat{\mathrm{I}}(x-1,y-1)$$

*où $\hat{I}(x,y) = 0$ si $x < 0$ ou $y < 0$.*

*À partir de cette image intégrale, la somme des valeurs dans une zone rectangulaire de l'image originale* $\mathrm{I}$ *peut être obtenue en utilisant les valeurs des 4 pixels dans l'image intégrale* $\hat{\mathrm{I}}$ *:*

$$\sum_{\substack{x_A < x' \leq x_C \\ y_A < y' \leq y_C}} \mathrm{I}(x',y') = \hat{\mathrm{I}}(A) + \hat{\mathrm{I}}(C) - \hat{\mathrm{I}}(B) - \hat{\mathrm{I}}(D)$$




1. *Implémenter votre propre code pour obtenir une image intégrale.*
2. *Re-implémenter le filtrage moyenneur de taille $N \times N$ en utilisant l'image intégrale.*
3. *Comparer le temps d'exécution pour les deux implémentations du filtrage moyenneur sur les différents tailles d'images et les différentes valeurs de $N$.*