

# Traitement d'image : Convolution

## 1 Première étape

Le but de ce TP est d'implémenter la convolution. C'est à dire de permettre d'appliquer un noyau (typiquement carré de largeur impaire) au voisinage de chaque pixel de l'image, et de convoluer les valeurs.

Le travail minimal demandé consiste à pouvoir appliquer une convolution pour une taille de fenêtre donnée (par exemple  $3 \times 3$ ), et de définir plusieurs opérateurs basés dessus. Ces opérateurs peuvent nécessiter d'appliquer un seul noyau (par exemple pour faire un filtre Moyenneur) ou plusieurs (par exemple pour faire un Extracteur de contour).

## 2 Généricité

La convolution pose plusieurs questions au niveau de l'architecture logicielle, afin d'éviter de dupliquer du code au maximum. Même si il n'est pas obligatoire d'y réfléchir pour le rendu de TP, ce sera nécessaire pour le projet. Plusieurs améliorations peuvent être envisagées :

- Permettre d'appliquer des filtres carrés de tailles quelconques.
- Permettre d'appliquer des filtres de tailles quelconques (utile par exemple pour les filtres séparables).
- Séparer l'itérateur sur le voisinage d'un pixel du calcul qui y est associé (ce qui permet de coder d'autres opérateurs défini sur un voisinage, tel que les filtres de rang).

## 3 Renderscript

Ces opérateurs sont assez coûteux en temps, et vous pourriez être tenté de vouloir améliorer les performances. Étant donné que les calculs sont indépendants, une implémentation multi-thread est particulièrement adaptée en général. Sous Android vous pourrez utiliser les threads, ou Renderscript qui sera plus efficace.

N'hésitez pas à regarder et vous comparer aux scripts intrinsic (<https://developer.android.com/reference/android/renderscript/ScriptIntrinsic.html>) qui permettent par exemple de faire une convolution  $3 \times 3$ ,  $5 \times 5$  ou un un filtre gaussien sur une image.