

# TP2 : Restauration d'images

## Objectif(s)

La restauration d'images consiste à restituer aussi fidèlement que possible le signal d'origine afin de réduire l'information inutile dans une image. Une des techniques les plus utilisées est la suppression du bruit qui s'effectue par filtrage de l'image.

## Exercice 1 : Suppression d'un bruit impulsif

Nous allons travailler sur l'image couleur `Rubik's_cube_R_impulse.png` contenant un bruit de type impulsif souvent appelé bruit poivre et sel.

1. Chargez et affichez l'image couleur pour voir quelle quantité de bruit est présente dans l'image. Essayez de débruiter cette image en appliquant un **filtre moyennneur** de taille **3x3** via la fonction `blur()`. Que constatez-vous ?  
Essayez d'appliquer à nouveau le **filtre moyennneur** de taille **3x3** sur l'image résultat. Est-ce que cela change le résultat ? Si oui, testez d'appliquer plusieurs fois ce **filtre**. Est-ce que le résultat converge vers une image acceptable ?  
Essayez maintenant de prendre un **filtre** de plus grande taille, genre **5x5** ou **7x7**. Est-ce mieux ?
2. Appliquez à présent un **filtre gaussien** via la fonction `GaussianBlur()`. Essayez différentes tailles de filtre et d'enchaîner plusieurs **filtrages successifs** pour tenter de supprimer le **bruit**. Que constatez-vous par rapport au **filtre moyennneur** ?
3. Testez le **filtre bilatéral** via la fonction `bilateralFilter()`. Attention à bien lire la documentation avant de vous lancer dans les tests de cette fonction. Enchaenez plusieurs **filtrages successifs**. Qu'arrivez-vous à **débruiter** avec cette fonction de **filtrage** ? Est-elle adaptée à votre problématique ? Si vous aviez regardé la documentation au préalable, auriez-vous choisi cette fonction pour tenter de débruiter ?
4. Appliquez à présent un **filtre median** via la fonction `medianBlur()`. Essayez différentes tailles de filtre et d'enchaîner plusieurs **filtrages successifs** pour tenter de supprimer le **bruit**. Que constatez-vous par rapport aux **filtres** testés précédemment ?
5. Réutilisez les fonctions du TP1 pour générer et afficher le graphique en bâtons de l'**histogramme global** de l'image avant **débruitage**. Que pouvez-vous constater sur cet **histogramme** ? Réutilisez intelligemment la fonction `floodFill` afin de récupérer le **masque** d'arrière-plan (pixels blancs) et de générer un nouvel **histogramme** sans ces pixels blancs de l'arrière-plan. Pour cela, vous allez rechercher un pixel blanc depuis le coin en haut à gauche dans un carré le plus petit possible avec un parcours comme dans l'exemple ci-dessous :

1	2	5	10
4	3	6	11
9	8	7	12
16	15	14	13

Quel changement pouvez-vous constater sur cet **histogramme** ?

Vous pouvez encore faire mieux pour évaluer ce masque... utilisez pour cela les **algorithmes de morphologie mathématique** afin de combler les trous de votre masque et récupérer ainsi **TOUS** les pixels blancs de l'arrière-plan... attention, la morpho-math va vous permettre de boucher tous les trous mais vous allez devoir ensuite supprimer les pixels de bruit qui étaient à l'intérieur de cette zone. Pour cela, utilisez une **fonction de seuillage** pour ne récupérer que les pixels blancs et fusionnez ces deux masques pour obtenir ce que vous souhaitez.

Quel changement pouvez-vous constater sur cet **histogramme** ?

- Proposez une nouvelle fonction de **filtrage**, proche du **filtrage median**, qui permettrait de mieux conserver les **contours** dans l'image. Codez-la et testez-la. Vous aurez besoin de la fonction `copyMakeBorder()` pour cloner une image en lui ajoutant un bord, de stocker le voisinage de pixels dans une structure de **vector** et d'utiliser les fonctions `push_back()`, `size()`, `sort()` et `clear()` sur cette structure afin de récupérer le pixel médian.

Le résultat est-il à la hauteur de vos espérances ? Quel changement pouvez-vous constater sur cet **histogramme** ?

- Et si vous réutilisiez plusieurs fois le **filtre median** de taille 3x3 et que vous régénériez l'**histogramme** final.

Quel changement pouvez-vous constater entre cet **histogramme** et celui généré à la question précédente ?

- Nous allons à présent travailler sur l'image couleur `Rubik's_cube_R_gaussian.png` contenant un **bruit de type gaussien** souvent appelé **bruit blanc gaussien**.

Testez les différents filtres précédents pour évaluer celui (ou ceux) qui fonctionne(nt) le mieux. Quelle est votre conclusion ?