

TD 8

« Statistiques locales et traitements locaux »

1. Programmation naïve

a. Image des écart-types locaux

Ecrire un programme calculant l'image $R'(i,j)$ des écart-types locaux. Pour chaque pixel, l'écart-type local est calculé dans une fenêtre de taille S selon la formule :

$$R'(i,j) = \sqrt{\left[\frac{1}{S^2} \times \sum_{k=-S/2}^{+S/2} \sum_{l=-S/2}^{+S/2} R^2(i+k, j+l) \right] - \left[\frac{1}{S^2} \times \sum_{k=-S/2}^{+S/2} \sum_{l=-S/2}^{+S/2} R(i+k, j+l) \right]^2}$$

```
...
#define          S = 51 ;          /* size of the statistics window */
...
double          mean;              /* current local mean */

...
/*****
/*****
/* PROCESSING SECTION */
/*****
/* Initialize the processed image */
/*****
for (ichannel=0; ichannel<channel_number; ichannel++)
{
    for (ili=S/2 ; ili<nliin - S/2; ili++)
    {
        for (ipx=S/2; ipx<npxin - S/2; ipx++)
        {

                } /* Loop on pixels */
        } /* Loop on lines */
    } /* Loop on channels */
}
```

b. Mesure de performances et complexité

Calculer le temps d'exécution pour les tailles S suivantes de la fenêtre statistique:

S	3	11	51	101
temps (s)				

Quelle est la complexité de l'algorithme pour une image $M \times N$ et une fenêtre statistique $S \times S$?

2. Programmation optimisée

a. Formule de récurrence selon les colonnes (ligne i fixée)

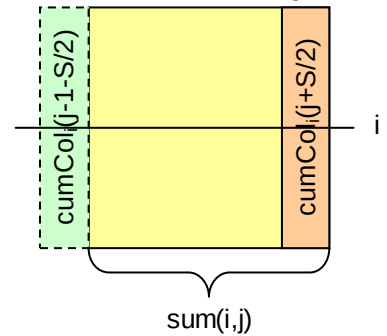
Soit $m(i,j)$ la moyenne locale calculée dans la fenêtre $S \times S$ centrée au point (i,j) .

$$m(i,j) = \frac{1}{S} \times \text{sum}(i,j) \quad \text{où } \text{sum}(i,j) \text{ est la somme des pixels dans la fenêtre statistique.}$$

Trouver l'expression récursive définissant $\text{sum}(i,j)$ en fonction d'elle-même et d'une somme $\text{sumCol}_i(j)$ des valeurs image selon les lignes au-dessus et en-dessous de la ligne i et dans la colonne j .

$$\text{sum}(i,j) = \text{sum}(i,j-1) \dots \dots \dots$$

Quelle est la complexité de l'algorithme avec cette première optimisation pour une image $M \times N$ et une fenêtre statistique $S \times S$?

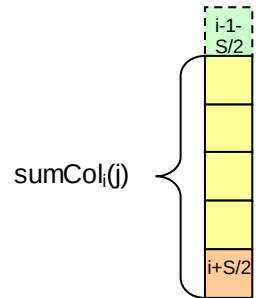


b. Formule de récurrence selon les lignes

Trouver l'expression récursive définissant la somme d'une colonne $\text{sumCol}_i(j)$ autour de la ligne i en fonction d'elle-même autour de la précédente ligne $(i-1)$ et de la valeur $R(i,j)$ du pixel dans l'image en entrée.

$$\text{sumCol}_i(j) = \text{sumCol}_{i-1}(j) \dots \dots \dots$$

Quelle est la complexité de l'algorithme avec cette seconde optimisation pour une image $M \times N$ et une fenêtre statistique $S \times S$?



3. Stretching linéaire imposé par les statistiques de sortie

Dans un traitement global, on cherche à réaliser le stretching linéaire ($R'(i,j) = A \times R(i,j) + B$) d'une image afin d'obtenir une image de sortie $R'(i,j)$ de moyenne m_0 et d'écart-type σ_0 choisis par l'utilisateur.

a. Stretching global - Expression mathématique

Exprimer les paramètres de stretching A et B en fonction de la moyenne m et de l'écart-type σ de l'image en entrée $R(i,j)$ et des valeurs de statistiques m_0 et σ_0 désirées en sortie.

b. Stretching local – Programmation optimisée

En utilisant la technique d'optimisation décrite au point 2, programmer l'application réalisant un stretching linéaire à partir de statistiques locales (moyenne $m(i,j)$ et écart-type $\sigma(i,j)$) estimées dans une fenêtre $S \times S$ centrée sur le pixel (i,j) et afin d'atteindre localement une moyenne m_0 et un écart-type σ_0 donnés par l'utilisateur.

c. Mesure de performances

Calculer le temps d'exécution de l'application optimisée pour les tailles S suivantes de la fenêtre statistique:

S	3	11	51	101
temps (s)				