

Compte rendu des Travaux Pratiques de la semaine



Imad Faouzi



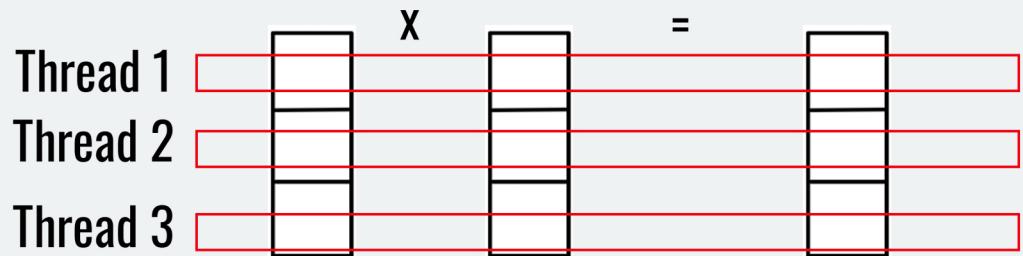
Fabrice Huet

Voici les fichiers et leur organisation.

No.	Travaux Pratiques	Fichier	Description
1	Multiplication de deux tableaux (A x B = C)	1D.ipynb	Réalisation de la multiplication de deux tableaux.
2	Conversion d'une image RGB en niveau de gris	grayFiltre 2D.ipynb	Transformation d'une image couleur (RGB) en niveau de gris.
3	Multiplication de deux matrices (A x B = C)	matrice 2D.ipynb	Effectuer la multiplication de deux matrices.
4	Flou gaussien sur la première version de l'image	Gaussian 1V.ipynb	Application d'un flou gaussien sur une première version de l'image avec la fonction "floute(image_source, image_destination)".
5	Flou gaussien sur la deuxième version de l'image	Gaussian 2V.ipynb	Application d'un flou gaussien sur une deuxième version de l'image avec la fonction "FlouteGaussienCustom(image_source, image_destination, largeur, sigma)".
6	Évaluation des performances avec différentes tailles de grille	Gaussian 3V.ipynb	Enregistrement du temps de début et de fin pour chaque taille de grille [5, 10, 14, 16, 20] et comparaison des résultats obtenus.



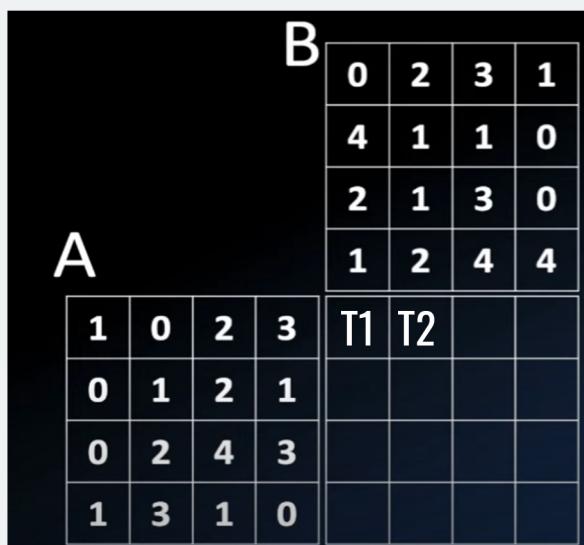
1 - la multiplication de deux vecteurs 1D :



Chaque thread va effectuer une seule opération pour un seul indice, par exemple, l'indice 0, où $A[0] \times B[0] = C[0]$.

2 - la multiplication de deux vecteurs 2D :

$$\begin{array}{|c|c|c|c|} \hline 1 & 0 & 2 & 3 \\ \hline 0 & 1 & 2 & 1 \\ \hline 0 & 2 & 4 & 3 \\ \hline 1 & 3 & 1 & 0 \\ \hline \end{array}
 \times
 \begin{array}{|c|c|c|c|} \hline 0 & 2 & 3 & 1 \\ \hline 4 & 1 & 1 & 0 \\ \hline 2 & 1 & 3 & 0 \\ \hline 1 & 2 & 4 & 4 \\ \hline \end{array}
 =
 \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}$$



Chaque thread va calculer un élément de la matrice résultante C en effectuant la multiplication de la ligne correspondante de la matrice A par la colonne correspondante de la matrice B.

3 - Prenez l'image dont le chemin est `image_source`, appliquez le flou avec votre kernel, et enregistrez le résultat sur le disque dans `image_destination`.

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

Dans cette première version, j'ai utilisé une matrice Gaussienne prédefinie de taille 5x5.



input



output

Explication :

Chaque thread va traiter un seul pixel de l'image. Il calculera la nouvelle valeur en considérant les deux voisins (dans ce cas particulier) de (-2, -2) jusqu'à (2, 2). Si une valeur n'existe pas, il prendra la valeur actuelle.

4 - Dans ce cas, nous allons générer un noyau gaussien avec les paramètres de largeur et de sigma, puis nous allons appliquer le flou à l'image. Ensuite, nous



Dans ce cas, la matrice gaussienne n'est pas constante et elle change en fonction de la taille et de sigma. Voici la fonction correspondante :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

5 - Dans la dernière version, nous allons évaluer les performances en fonction de la taille de la grille (thread). Pour ce faire, nous allons stocker le temps de début et de fin pour chaque taille [5, 10, 14, 16, 20]. En fin de compte, nous comparerons les résultats obtenus