

ASR TD5

Threads et calcul parallèle

1 Réduction

Effectuer la somme des éléments d'un tableau en utilisant plusieurs threads. Chaque thread effectue la somme des éléments d'un sous-tableau. Ces sommes locales doivent ensuite être cumulées par le thread principal.

2 Application de filtres

Dans les exercices suivants, il est demandé de réaliser chaque programme de manière séquentielle, puis en utilisant les threads. Il conviendra de vérifier que les résultats obtenus sont identiques. Les temps d'exécution seront mesurés et comparés à l'aide de la fonction `gettimeofday` (lire le man pour savoir comment utiliser la fonction!). Réutiliser la bibliothèque DevIL utilisée en TP pour la lecture et l'enregistrement des images.

2.1 Inversion des couleurs

Les couleurs comportent généralement 3 composants rouge, vert, bleu (RGB : Red, Green, Blue) codées chacune sur 1 octet (unsigned char). La valeur de chaque composante est donc comprise entre 0 et 255. Calculer le complément à 255 de chaque composante de manière à inverser les couleurs.

2.2 Niveaux de gris

Plusieurs techniques existent pour convertir une image en niveaux de gris. Nous utilisons ici une technique permettant de conserver la luminance de l'image de base (<http://en.wikipedia.org/wiki/Grayscale>) qui consiste à multiplier les composantes d'un pixel par différents coefficients, à additionner ces valeurs, puis à les ramener dans le domaine $[0, 255]$ suivant la formule suivante :

$$out.R = out.G = out.B = (in.R \times 307 + in.G \times 604 + in.B \times 113) / 1024$$

Attention au stockage des valeurs intermédiaires qui risquent de ne pas tenir sur 1 octet.

2.3 Flou

Plusieurs techniques existent pour calculer un flou, nous utiliserons ici celle qui consiste à faire la moyenne de la valeur du pixel courant avec celles de ses 4 voisins et qui peut se traduire par la matrice de convolution suivante :

0	1	0
1	1	1
0	1	0

On applique le coefficient 1 au pixel courant (centre de la matrice), et aux pixels voisins (dessus, dessous, gauche, droite), on effectue la somme des pixels (pour chaque composante) puis on divise par le nombre de pixels considérés, ici 5. Voici la formule à appliquer pour la composante rouge de chaque pixel :

$$out.R = (in.R + up.R + down.R + left.R + right.R) / 5$$

2.4 Autres filtres

Vous pouvez tester d'autres filtres en appliquant d'autres coefficients. Quelques exemples sont disponibles à l'adresse suivante : <https://docs.gimp.org/fr/gimp-filter-convolution-matrix.html>

3 Tri

Effectuer le tri d'un tableau en utilisant plusieurs threads. Une solution simple consiste à trier des sous-parties du tableau, puis de fusionner ces sous-tableaux. Commencer par utiliser 2 threads. Comment effectuer la fusion si on utilise 4 threads ?