

Travaux Pratique 8

Ce sujet est en lien avec le quatrième chapitre du cours, et concerne la programmation CUDA. Les mêmes commentaires que ceux des derniers TP s'appliquent ici aussi.

En imagerie numérique, l'égalisation d'histogramme est une méthode d'ajustement du contraste d'une image donnée (cf. http://en.wikipedia.org/wiki/Histogram_equalization). Pour une image en niveaux de gris, l'idée est de calculer un histogramme comptant l'utilisation de chaque niveau de gris, de calculer la fonction de répartition de cet histogramme, puis d'étaler les niveaux de gris utilisés.

Plus précisément, soit $\{x_i\}$ l'ensemble des pixels d'une image définie sur L niveaux de gris. L'histogramme est un tableau comptant les occurrences de chaque niveau de gris noté l , pour $l \in [0 \dots L - 1]$:

$$h(l) = \sum_{i=0}^{n-1} \delta(x_i - l),$$

où n est le nombre de pixels de l'image, et δ est la fonction de Dirac telle que :

$$\delta(\xi) = \begin{cases} 1 & \text{si } \xi = 0, \\ 0 & \text{sinon.} \end{cases}$$

La fonction de répartition r (*cdf* en anglais) est définie sur l'intervalle des niveaux de gris comme la somme des nombres d'occurrence des valeurs précédentes :

$$r(l) = \sum_{k=0}^{L-1} h(k).$$

Eh oui, c'est une somme préfixe, donc un SCAN inclusif ! La transformation suivante permet « d'étaler » l'histogramme (HE pour *histogram equalization*), en transformant en densité de probabilité d'abord, puis en mettant à l'échelle du domaine des niveaux de gris :

$$T(x_i) = (L - 1) \frac{r(x_i)}{n}.$$

Cette méthode est étendue aux images couleurs en appliquant cette transformation sur la composante « intensité » (V) de la couleur exprimée dans le repère HSV : *Hue* (Teinte), *Saturation* et *Value* (cf. http://en.wikipedia.org/wiki/HSL_and_HSV). Avec des images 24 bits, la valeur s'exprime sur un seul octet ; donc L vaut 256 (et $L - 1 = 255$).

Nous allons jouer avec l'implantation de l'histogramme et le scan inclusif.

ATTENTION : pour rappel, tous les travaux non rendus sont transformés en note 0. Si un tiers ou plus des TP ne sont pas rendus (à partir de 4 TP), la note attribuée à l'épreuve sera ABI sans contrôle de rattrapage, conduisant de fait à l'échec au module et à l'année.

Votre travail est à rendre (il le sera à chaque fois) sous la forme du code (et uniquement la partie « student ») accompagnée d'un rapport au format PDF, le tout dans une archive compressée au format ZIP. Ne respecter pas ces contraintes, et votre note en sera diminuée de quelques points.

Vous ne devez modifier que ce qui se trouve dans le répertoire « ./student/ » !

Notez que chaque exercice vient avec un squelette, qui s'occupe de la ligne de commande, du lancement de votre code (défini dans une classe particulière), et d'une vérification sommaire du résultat (lorsque c'est possible). Utilisez l'option « -h » ou « --help » pour connaître le fonctionnement de la ligne de commande ...

Exercice 1

Implémentez la fonction `rgb2hsv` qui, pour chaque pixel de l'image, calcule sa valeur dans l'espace HSV en utilisant la fonction `RGB2HSV`, et **répartit le résultat dans trois tableaux** différents. Notez qu'il s'agit d'une forme particulière de `SCATTER`. Ce type de répartition en trois tableaux vise à optimiser le débit mémoire d'un kernel CUDA (encore la coalescence).

Implémentez la transformation inverse (`hsv2rgb`), de HSV vers RGB, en utilisant la fonction `HSV2RGB`.

Exercice 2

Calculez l'histogramme $h(l)$ des valeurs. Si les valeurs reçues sont en `float`, elles restent comprises entre 0 et 255.

Exercice 3

Calculez la fonction de répartition $r(l)$.

Exercice 4

Calculez la transformation finale $T(x_i)$.

Exercice 5

Implantez la version AHE décrite dans la section III.A, équation 6, de l'article suivant :

T. Arici, S. Dikbas, and Y. Altunbasak, [A histogram modification framework and its application for image contrast enhancement](#), IEEE Trans. Image Process., vol. 18, no. 9, pp. 1921–1935, Sep. 2009.

Expliquez les choix ayant conduits à votre implantation de cette méthode.

Notez ici que la transformation de l'histogramme, si elle était directement implantée avant le calcul de la fonction de répartition en utilisant des entiers, donnerait des erreurs numériques importantes. Pour éviter cela, il vous est demandé d'appliquer l'équation (6) à la volée lors de la transformation finale de la valeur V de chaque pixel. Cela est rendu possible car l'équation (6) n'est ni plus ni moins qu'une application linéaire (et donc s'applique aussi bien sur les deux fonctions de répartition résultant des deux histogrammes h et u).

Notez enfin que pour $\lambda = 0$ vous retrouvez l'algorithme HE...

Exercice 6

Implantez la version WHE décrite section III.C de l'article vu dans l'exercice précédent (équation 12). Notez ici que dans l'article la phrase « *the average local variance of pixels with gray-level i* », doit être lue pour des images en couleur en remplaçant « niveau de gris » par « valeur ».

Une erreur classique sur le calcul de la variance est de considérer comme correcte la formule suivante :

$$\sigma^2 = \frac{1}{n} \sum (x_i - \bar{X})^2$$

où \bar{X} est la moyenne empirique :

$$\bar{X} = \frac{1}{n} \sum x_i.$$

Cette formulation de la variance est correcte en théorie, mais en utilisant des données réelles (comme le voisinage d'un pixel) alors cette formule est celle dite de la [variance empirique](#). Or, cette variance empirique est biaisée... Pour corriger ce défaut, il faut utiliser la formulation suivante :

$$\widetilde{\sigma^2} = \frac{n}{n-1} \sigma^2 = \frac{1}{n-1} \sum (x_i - \bar{X})^2.$$

Une petite subtilité concerne encore le calcul de la variance, celle locale aux pixels de chaque valeur V dans l'algorithme AHE. Sachant que ces variances seront ensuite moyennées pour

chaque valeur V , une possibilité est de la calculer localement avec deux pixels seulement : le pixel X en position (row, col) de valeur V , et le pixel Y en position $(row, col + 2)$ par exemple et de valeur V_y . Ce procédé évite un calcul coûteux dans une fenêtre autour de X qui serait d'une taille à définir. Cette solution explique pourquoi la fonction à coder ne reçoit pas de taille de fenêtre de calcul de la variance locale.

Dans le rapport, expliquez le calcul de la variance locale que vous effectuez.

Notez ici que la fonction à coder doit réaliser tout le traitement à appliquer sur la valeur V de chaque pixel : d'abord le calcul de l'histogramme et de la moyenne des variances locales, puis la transformation de l'histogramme, le calcul de la fonction de répartition et enfin l'application de la transformation...

Résultats (HE)

Image "Fragornard,_The_Swing.ppm"



Image « Roy_Lichtenstein_Drowning_Girl.ppm »



Image “Las_Meninas,_by_Diego_Velasquez,_from_Prado_in_Google_Earth.ppm”



Image “Caravaggio_-_La_vocazione_di_San_Matteo.ppm »

