

Devoir d'informatique

Consignes

- Le devoir se fera sur copie double uniquement.
- Le numéro de chaque exercice et de chaque question devra être indiqué sur votre copie.
- Les indentations devront correctement figurer sur votre copie. Vous pourrez par exemple tracer une barre verticale.
- Pour chaque fonction vous donnerez au plus une ligne de commentaire permettant de spécifier votre fonction.

1 Représentation des nombres

On dispose d'un système d'exploitation où les nombres sont représentables sur 10 bits.

Question 1 Combien de nombres entiers peut-on représenter ?

Question 2 Quels sont le plus petit et le plus grand entier naturel représentables ?

Question 3 Quels sont le plus petit et le plus grand entier relatif représentables ?

Question 4 Convertir en base 2 le nombre suivant : $(155)_{10}$.

Question 5 Convertir en base 16 le nombre suivant : $(155)_{10}$.

Question 6 Convertir en base 10 le nombre suivant : $(B3)_{16}$.

On dispose d'un système d'exploitation où les nombres sont représentables sur 4 bits.

Question 7 Convertir en base 2 le nombre suivant : $(-6)_{10}$.

2 Mesure du contraste

Le principe de cette méthode repose sur le fait qu'une image bien mise au point présente un maximum de contraste. Plus précisément l'écart de valeur entre les pixels est la plus grande. En fonctionnement, l'appareil mesure le contraste, puis par tâtonnement, va déplacer l'objectif jusqu'à détecter le maximum.

Représentation d'une image

Une image matricielle est représentée sous la forme d'une matrice. Chaque élément de la matrice correspondra à un pixel qui est généralement représentée par une liste de 3 entiers naturels, représentant les composantes rouge, vert et bleu; c'est le codage RVB.

Chaque valeur est représentée par un entier allant de 0 à 255. On suppose, dans cette partie, travailler avec un appareil photo doté d'un capteur de 48 MPixels, c'est-à-dire que la photo sera composée de 48 millions de pixel.

On suppose que le stockage des photos se fait en mode RAW sans compression, c'est-à-dire que l'on stocke directement ce que récupère le capteur.

L'image sera représentée en mémoire dans une variable I , de type liste de listes Python classique ou de type `numpy.ndarray`. Dans les deux cas, on accédera aux données RVB d'un pixel de coordonnées (i, j) , par l'instruction `I[i][j]`. Vous choisirez le type que vous préférez manipuler.

Question 8 Préciser l'espace mémoire nécessaire pour stocker la valeur d'une composante, puis celle d'un pixel et enfin celle d'une image en Mo (= 1000 ko) ou Mio (= 1024 ko).

Conversion en niveau de gris

La première étape de la détection du contraste est de convertir l'image en niveau de gris pour n'obtenir qu'une valeur par pixel.

Cette transformation s'opère en plusieurs étapes :

- chaque composante R, V et B qui ne sont pas linéaires en terme d'intensité lumineuse sur le rendu, sont d'abord transformées dans un espace linéaire. Une composante C sera transformée selon la définition suivante :

$$- C_{lin} = \frac{C}{12,92} \text{ si } C \leq 0,0404,$$

$$- C_{lin} = \left(\frac{C + 0,055}{1,055} \right)^{2,4} \text{ sinon ;}$$

- la valeur du niveau de gris Y_{lin} en échelle linéaire sera calculée à partir des valeurs linéarisées par : $Y_{lin} = 0,2126R_{lin} + 0,7152V_{lin} + 0,0722B_{lin}$;
- on repasse dans l'espace non linéaire avec le calcul suivant :
 - $Y = 12,92 Y_{lin}$ si $Y_{lin} \leq 0,0031308$;
 - $Y = 1,055 Y_{lin}^{1/2,4} - 0,055$ sinon.

Question 9 Écrire une fonction `Clinear(val)`, qui prend en argument une valeur de l'espace non linéaire et qui renvoie la valeur linéarisée.

Question 10 Écrire une fonction `Y(pix)` qui prend en argument une liste de trois valeurs correspondant à un pixel au format RVB et qui renvoie la valeur Y du niveau de gris dans l'espace non linéaire.

Question 11 Écrire une fonction `NiveauxGris(I)` prenant en argument une image I au format RVB et qui renvoie une image de même dimension en niveau de gris.

Pour détecter le contraste d'une image en niveau de gris, on va comparer pour chaque pixel les valeurs autour de celui-ci. Plus l'écart est grand (ce qui est le cas quand l'image est nette), plus les pixels autour du pixel de référence ont une valeur différente ; on calcule d'une certaine manière la dérivée en chaque pixel.

Cette opération est réalisée par un filtre de Sobel. Partant d'une image I , on extrait les pixels autour du pixel (i, j)

sous la forme d'une matrice 3x3, notée $I_e = \begin{pmatrix} I_{i-1,j-1} & I_{i-1,j} & I_{i-1,j+1} \\ I_{i,j-1} & I_{i,j} & I_{i,j+1} \\ I_{i+1,j-1} & I_{i+1,j} & I_{i+1,j+1} \end{pmatrix}$.

On réalise ensuite une convolution entre cette matrice et la matrice de filtration. On définit la convolution entre deux matrices A et B de taille 3x3 par $A \otimes B = \sum_{i=0}^2 \sum_{j=0}^2 A_{ij} B_{ij}$.

On réalise une filtration selon les deux directions de l'image. On donne les deux matrices de filtration :

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ et } G_y = \begin{pmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Enfin on calcule une norme euclidienne des deux convolutions dont on prend la partie entière pour obtenir la valeur du contraste c du pixel (i, j) : $c = \text{Ent} \left(\sqrt{(I_e \otimes G_x)^2 + (I_e \otimes G_y)^2} \right)$.

Pour terminer, on calcule le contraste de cette manière pour chaque pixel intérieur à l'image (on ne calcule rien sur les bords pour simplifier), puis on réalise la moyenne des valeurs pour obtenir l'indice de contraste de référence c_{ref} .

Question 12 Écrire une fonction `convolution(A,B)` prenant en argument deux matrices de taille 3x3 et qui renvoie la valeur du produit de convolution.

Question 13 Écrire une fonction `contraste_pixel(I,i,j)` prenant en argument une image I au format niveaux de gris et les coordonnées du pixel (i, j) qui renvoie la valeur du contraste défini précédemment par la quantité c .

Question 14 Écrire une fonction `contraste(I)` prenant en argument une image `I` au format niveau de gris et qui renvoie la valeur du contraste de référence c_{ref} .

Pour régler la netteté de l'image, l'objectif est déplacé à l'aide d'un moteur pas à pas. L'objectif est déplacé d'un pas, une photo est prise, la valeur du contraste de référence est calculée puis comparée à la valeur obtenue au pas précédent; l'algorithme s'arrête dès que la valeur du contraste de référence diminue. Le moteur pas à pas recule d'un pas pour retourner à la position précédente où le contraste était maximale.

Pour cela, on utilise une fonction `position_objectif(val)` qui prend en argument un entier `val` allant de 0 à 1000 correspondant à la position en pas demandée et qui déplace l'objectif à cette position. Une fonction `prise()` permet de prendre un cliché et de retourner une image au format RGB.

Question 15 Écrire une fonction `reglage`, dont les arguments et les valeurs de retour sont à définir, répondant au comportement décrit en partant de la position 0 en pas. On supposera que le maximum de contraste existe.