Manipulations élémentaires d'images couleurs

Avertissement

- Cette étude nécessite la boîte à outils `imlab`. Pour y avoir accès, il faut exécuter la commande
  `exec directory/imlab2.1/loader.sce`
  où `directory` désigne le répertoire où est installé `imlab`, typiquement
  `/usr/local/scilab-4.1.1/contrib`
  Par la suite, vous pourrez insérer cette ligne dans un fichier `.scilab`. Ce dernier est exécuté dès le démarrage du logiciel (si le fichier est présent dans votre répertoire de connexion).
- Notez que l'instruction `exec` peut être utilisée, plus généralement, pour lancer tout script Scilab (suite de commandes Scilab placées dans un fichier dont le nom se termine par l'extension `.sce`).
- Scilab peut éventuellement manquer d'espace mémoire pour certaines opérations. Il suffit alors d'étendre la taille de la pile mémoire prédéfinie, en tapant par exemple
  `stacksize(10000000);`

Travail à réaliser

1. A l'aide de la fonction `imread`, charger l'image `cezannetm.ppm` (récupérable sur le Web). Quelle est la dimension de cette image ? La visualiser avec la commande `imshow`.
2. Afficher les composantes R, G et B de cette image.
3. En fait, comme vous pouvez l'observer, ces composantes ont été "mélangées" par un vilain plaisantin. Retrouver l'ordre naturel en essayant les différentes permutations possibles. Désormais, on travaillera sur l'image dont les composantes ont été réordonnées.
4. Que valent les valeurs des 3 composantes pour le pixel (448,98) ? On peut obtenir cette information soit directement par inspection des valeurs, soit manuellement à l'aide de la souris. Tester les deux méthodes. Effectuer un zoom sur une partie apparemment uniforme de l'image. Que remarque-t-on ?
5. Comment faire pour générer une image binaire où l'on met à 1 les valeurs pour lesquelles la composante R est prépondérante ? (Utiliser la commande `find`.) Réaliser la même opération pour les composantes G et B.
6. De nombreux formats existent pour la représentation des images couleur. Un format particulier, employé notamment en vidéo, est le format YUV : Y s'interprète comme une composante de luminance alors que U et V correspondent à des composantes de chrominance. Le lien entre les composantes R, G, B et Y, U, V est donné par la relation matricielle suivante :

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.146 & -0.288 & 0.434 \\ 0.617 & -0.517 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

   En utilisant cette relation, calculer les composantes YUV de l'image étudiée. Visualiser ces composantes. Quels sont, selon vous, les avantages de la représentation

YUV par rapport à la représentation RGB ? Que valent les valeurs minimales et maximales des composantes Y, U et V ? Quelles sont les valeurs théoriques pouvant être atteintes ?

7. Effectuer l'opération inverse pour revenir dans le domaine RGB. Si l'on arrondit les composantes Y, U et V à l'entier le plus proche, que vaut l'erreur quadratique moyenne de reconstruction ?
8. Pour éviter l'erreur de troncature précédente, la nouvelle norme de compression JPEG 2000 pévoit l'emploi d'une transformation "integer to integer" permettant de conserver le caractère entier des composantes transformées. Cette transformation est définie par :

$$\begin{aligned}\widetilde{Y} &= \lfloor (R + 2G + B)/4 \rceil \\ \widetilde{U} &= B - G \\ \widetilde{V} &= R - G\end{aligned}$$

où $\lfloor x \rceil$ désigne la valeur de x arrondie au plus proche entier. (On utilisera sous Scilab, l'instruction `floor(x+0.5)`.) Mettre en œuvre cette tranformation sur l'image traitée.

9. Vérifier que cette transformation peut être inversée, *sans aucune perte d'information*, à l'aide des formules suivantes :

$$\begin{aligned}R &= \widetilde{Y} + \widetilde{V} - \lfloor (\widetilde{U} + \widetilde{V})/4 \rceil \\ G &= \widetilde{Y} - \lfloor (\widetilde{U} + \widetilde{V})/4 \rfloor \\ B &= \widetilde{Y} + \widetilde{U} - \lfloor (\widetilde{U} + \widetilde{V})/4 \rceil.\end{aligned}$$