

## IMA3 - Projet de programmation structurée 2016-2017

### Toolkit image

Le projet qui vous est proposé vise à mettre en place un certain nombre d'opérations élémentaires d'images. Une base logicielle vous est fournie avec ce sujet (<http://www.cristal.univ-lille.fr/~grisoni/IMA/image.c>), proposant une implémentation des fonctions de chargement et de sauvegardes d'images couleur. Le fichier <http://www.cristal.univ-lille.fr/~grisoni/IMA/Lena.512.ppm> est un exemple d'image 512x512 dans le format d'image utilisé, qui pourra vous servir de base de travail. Le format d'image qui est proposé est le format de stockage d'image PPM (un format couleur standard, utilisable par tous les logiciels de visualisation d'image, ayant de surcroît l'avantage d'être très simple). La base de code vous fournit un petit exemple d'utilisation des fonctions de chargement et de sauvegarde.

Vous devez envoyer par email à votre encadrant direct de TP une archive de votre travail, **incluant un rapport court** (5 pages environ, on ne demande pas d'inclusion de code dans ce rapport), pour le **16 Janvier 2017 à 17H. Passé cet horaire, une pénalité de retard pourra être appliquée.**

Une image sera stockée par votre programme dans un tableau à 2 dimensions, chaque pixel étant représenté par trois octets, un pour chaque composante de couleur affichée (Rouge, Vert, Bleu). **VOUS N'AVEZ PAS BESOIN D'UTILISER DE POINTEUR POUR REALISER CE PROJET.** Toutefois, vous pouvez le faire si vous le souhaitez. On demande les fonctionnalités suivantes:

- 1) **recopie d'image:** Pour cette opération on demandera à l'utilisateur le nom du fichier contenant l'image à charger, et le nom du nouveau fichier dans laquelle l'image sera stockée.
- 2) **Redimensionnement d'image:** l'image chargée est remplacée par une image de dimension deux fois inférieure (sur les deux dimensions, elle comprend donc au total quatre fois moins de pixels).
- 3) **Flou:** On mettra en place une technique consistant à remplacer chaque pixel de l'image par une moyenne des pixels sur un voisinage local; plus la taille du voisinage est large, plus le résultat sera flou. On pourra appliquer cette opération plusieurs fois; ceci, dans le cas de voisinage de petite taille, permet d'obtenir un contrôle intéressant sur le résultat obtenu. l'implémentation de la fonction demandera à l'utilisateur la taille de la fenêtre de moyennage, ainsi que le nombre de passe à effectuer (le nombre de fois ou le flou sera appliqué).
- 4) **passage de l'image du format RGB vers le format TSV.** Le format TSV est un format de représentation qui pourra être intéressant pour les opérations de manipulation de couleur. Vous trouverez ci-contre les expressions pour passer, pour un pixel, de sa représentation RGB à sa représentation TSV. Il s'agit d'un codage de la couleur qui est plus proche du comportement de la vision humaine (Deux couleurs « proches » au format TSV seront vues comme semblable par l'oeil humain, ce qui n'est pas forcément le cas en représentation RGB). Les trois composantes sont les suivantes : la teinte  $T$  définit la couleur « pure », il s'agit

$$\begin{aligned}
 m &= \min(r, v, b) \\
 M &= \max(r, v, b) \\
 t &= \begin{cases} 0 & \text{si } M = m \\ 60 \frac{v-b}{M-m} + 360 & \text{si } M = r \\ 60 \frac{b-r}{M-m} + 120 & \text{si } M = v \\ 60 \frac{r-v}{M-m} + 240 & \text{si } M = b \end{cases} \\
 s &= \begin{cases} 0 \\ \frac{M-m}{M} \end{cases} \\
 v &= M
 \end{aligned}$$

d'un angle compris entre 0 et 360°. La saturation  $S$  représente l'intensité de la couleur, et enfin la valeur  $V$  est la luminosité. Ce modèle correspond à l'*approche du peintre*, voulant à partir de tube de peinture faire un mélange aboutissant à une couleur précise:  $V$  représente le niveau de gris dont il a besoin (le ratio de mélange entre blanc et noir),  $T$  la teinte de couleur pure, et  $S$  le ratio de mélange à faire entre  $T$  et  $V$  pour obtenir la couleur désirée. Il est à noter que cette représentation n'est pas destinée à être sauvegardée sur disque, le format d'image considéré ne pouvant le prendre en charge.

- 5) **Passage du TSV au RGB.** Transformation inverse de la précédente, utile si on a utilisé l'espace TSV pour manipuler l'image, et que l'on s'apprete à sauvegarder. Les équations sont les suivantes.
  - On pose  $N = E(T/60) \text{ modulo } 6$ ; //  $E()$  désigne ici la partie entière
  - On pose  $F = T/60 - E(T/60)$
  - On pose  $P = V*(1-S)$
  - On pose  $Q = V*(1-F*S)$
  - On pose  $U = V*(1-(1-F)*S)$
  - $(R,G,B)$  vaut, suivant que  $N$  vaut 0,1,2,3,4,5, ou 6:  $(V,U,P)$ ,  $(Q,V,P)$ ,  $(P,V,U)$ ,  $(P,Q,V)$ ,  $(U,P,V)$ ,  $(V,P,Q)$
- 6) **Extrapolation d'image:** cette fonctionnalité calcule l'image résultat de l'interpolation linéaire entre l'image chargée et l'image *moyenne*, cette dernière étant l'image dont tous les pixels valent la valeur moyenne de l'image (le  $R$  de chaque pixel stocke la valeur moyenne de tous les  $R$  de l'image, et pareil pour les  $G$  et  $B$ ). Pour cette extrapolation, on demandera à l'utilisateur un coefficient  $u$  qui sera utilisé pour l'interpolation linéaire de l'image. Pour chaque pixel, si on considère  $P$  comme étant la valeur initiale, et *moy* la valeur moyenne, alors la nouvelle valeur du pixel vaut :  $u*P + (1-u)*\text{moy}$ . le coefficient  $u$  est, pour l'interpolation standard, à fixer entre 0 et 1, mais on remarquera que l'on peut utiliser l'expression pour des valeurs supérieures à 1, ou inférieures à 0, également. On pourra faire quelque test pour explorer les effets de couleur que cette fonction permet.
- 7) Vous pouvez laisser libre cours à votre imagination pour rajouter d'autres fonctions de manipulation. Cela sera pris en compte dans la notation, en bonus, en fonction de ce que vous avez proposé. N'oubliez pas de décrire ces fonctions dans votre rapport si vous en implémentés !
- 8) **Menu d'application :** votre programme, utilisable en ligne de commande, permettra à l'utilisateur d'utiliser les fonctionnalités précédentes. Il vous permettra notamment de tester les différentes fonctions implémentées, et ne prendra pas en charge la visualisation proprement dite des images, celles-ci pouvant être visionnées par un outil tiers, présent sur vos machines de TP. Vous pourrez notamment valider votre programme en cherchant à mettre en place une image similaire à celle ci-dessous (vous pourrez inclure votre plus bel exemple dans votre rapport de projet !)



Marylin Monroe - Andy Warhol 1962