
Programmation Impérative

TP Tableaux 2D : manipulation d'images

1) Mise en place

- Assurez vous que vous disposez du support de cours sur les tableaux 2D et les manipulations élémentaires d'image, disponible sur pedago-etu.ensiie.fr/pub/C-TAB2D :
TP-IPI-Image-CM-2017.pdf
- Récupérez l'archive, aussi sur pedago-etu.ensiie.fr/pub/C-TAB2D :
TP-IPI-Image-Kit.zip
- Décompressez l'archive en exécutant dans le terminal la commande :
\$ unzip TP-IPI-Image-Kit.zip
- Compilez et exécutez pour vous assurer que tout est en ordre (la première image est l'image à traiter, la deuxième est créée par le programme, c'est le résultat du traitement) . Dans le terminal exécutez :
\$ cd tp-image
\$ make
\$./tp_image test.pgm res.pgm
et affichez les images test.pgm et res.pgm, par exemple comme ceci :
\$ atril test.pgm res.pgm

2) Travail à réaliser

Vous devrez rajouter des fonctionnalités à l'application dont le squelette vous a été fourni, et qui ne comporte initialement que quelques fonctions de transformation d'images en noir et blanc (1 octet par pixel) : inversion vidéo (négatif), ajout de bruit (dans le but de tester 2 approches de correction de défauts d'images), un effet fish-eye et l'élimination d'une ligne sur 2. Vous devrez aussi modifier le programme principal afin qu'il fasse appel aux nouvelles fonctions que vous avez mises en oeuvre. Pour recopier votre application il suffira de relancer la commande make dans le terminal, et de relancer l'exécution comme précédemment.

Fonctions à intégrer dans le module effets-photom.c

- Fonction d'éclaircissement d'image : `plusClairNB`. Cette fonction doit rendre votre image plus claire en augmentant chaque niveau de gris de 50%.
- Fonction de création d'un effet de relief : `deriv1xNB`, chaque pixel est obtenu en effectuant la différence entre son niveau de gris et celui du pixel qui se situe immédiatement à sa gauche. La différence obtenue peut prendre des valeurs entre -255 et 255. On tronquera les valeurs en dehors de l'intervalle [-128,127], qu'on ramènera entre 0 et 255, simplement en ajoutant 128.

Fonctions à intégrer dans le module effets-geom.c

- Fonction de création d'un effet de « fonte » : `effetFonteNB`. Cette fonction traite N pixels choisis aléatoirement. En général on prend N égal au nombre de pixels. Chaque pixel sélectionné (certains le seront plusieurs fois) est comparé à celui situé immédiatement sur la ligne inférieure. Si le pixel du bas est plus clair, alors il prend la valeur du pixel au dessus.
- Fonction de réduction de la taille d'une image : `quartImageNB`. Cette fonction calcule une nouvelle image dont les dimensions sont les dimensions initiales divisées par 2, et les pixels sont obtenus en ne retenant qu'une ligne sur 2, et sur chaque ligne, une colonne sur 2.

Fonctions à intégrer dans le module filtrage.c

- Fonction de lissage d'une image par calcul de la moyenne des pixels sur un voisinage : `filtrerImageNB`. Cette fonction a pour but de remplacer la valeur d'un pixel par la moyenne des valeurs de ses voisins pris dans une fenêtre carrée de taille impaire centrée sur le pixel à filtrer. Pour vous faciliter la mise au point décomposez votre problème en sous problèmes et donc écrivez des fonctions intermédiaires. Faites attention à ce que le filtrage soit applicable, ce n'est pas le cas des pixels proches du bord de l'image. Ajoutez les commentaires au format Doxygen et mettez à jour le fichier d'entête `filtrages.h`

Tests de vos fonctions dans le module main.c

- Vous modifierez le module `main.c` pour qu'il propose un menu avec les différentes fonctions disponibles.