L2 Informatique – AII3D – Traitement et d'analyse d'images
Programmation d'opérateurs en langage C-ANSI

Alain Crouzil

1 Récupération des outils et des données

Récupérez les archives `limace-1.6.zip`, `tai-2.zip` et `donnees.zip` sur le wiki enseignement.

L'archive `limace-1.6.zip` contient les fichiers de la bibliothèque LIMACE :

- `limace.c` : fichier source ;
- `limace.h` : fichier des en-têtes commentés ;
- `limace.pdf` : manuel utilisateur.

L'archive `tai-2.zip` contient un embryon de bibliothèque TAI d'opérateurs de traitement et d'analyse d'images que vous allez compléter avec d'autres opérateurs. Elle contient les fichiers suivants :

- `tai.c` : fichier source de la bibliothèque ;
- `tai.h` : fichier des en-têtes des fonctions de la bibliothèque ;
- `inversion.c` : code source de l'opérateur d'inversion d'une image ;
- `hist2im.c` : code source de l'opérateur de création d'une image représentant un histogramme ;
- `Makefile` : fichier de description des dépendances permettant de compiler les opérateurs.

L'archive `donnees.zip` contient des données permettant de tester vos opérateurs :

- `lena.pgm` : image de niveaux de gris au format PGM ;
- `histlena.mat` : fichier au format `Matrix` (LIMACE) contenant l'histogramme de l'image `lena.pgm` ;
- `lut-inv.mat` : LUT d'inversion d'une image stockée dans un fichier au format `Matrix` ;
- `moy3x3.mat` : masque, de taille 3×3, de lissage par le filtre moyenne stocké dans un fichier au format `Matrix` ;
- `sobelh.mat` : masque 3×3 de Sobel permettant de calculer la composante horizontale du vecteur gradient de chaque pixel ;
- `sobelv.mat` : masque 3×3 de Sobel permettant de calculer la composante verticale du vecteur gradient de chaque pixel.
- `vaisseaux.pgm` : image de niveaux de gris au format PGM qui permet de mettre en évidence les effets de l'égalisation d'histogramme ;
- `disques-et-tiges.pbm`, `im-bin.pbm`, `res-ero.pbm` et `res-dil.pbm` : images binaires au format PBM ASCII (attention, si vous affichez le contenu de ces fichiers, avec ce format, 1 représente le noir et 0 représente le blanc) ;
- `cel.pgm` : image de niveaux de gris au format PGM qui montre une coupe de tissu cérébral contenant des cellules nerveuses (grandes tâches grises avec un noyau circulaire plus sombre au milieu) et des cellules gliales (petits cercles noirs isolés) ;
- `enveloppe.pgm` : image de niveaux de gris au format PGM pouvant servir à tester la binarisation par seuillage ;
- `carre3x3.mat` : fichier au format `Matrix` (LIMACE) contenant un élément structurant ayant la forme d'un carré ;
- `disque7x7.mat` et `disque11x11.mat` : fichiers au format `Matrix` (LIMACE) contenant des éléments structurants ayant la forme d'un disque.

2 Prise en main des outils

1. Lisez attentivement le manuel utilisateur de LIMACE.
2. Examinez attentivement le contenu des fichiers constituant la bibliothèque TAI.

3. Compilez les programmes en utilisant la commande `make`.
4. Testez l'opérateur d'inversion sur l'image `lena.pgm` et l'opérateur de visualisation d'un histogramme sur l'histogramme `histlena.mat`.

3 Programmation d'opérateurs

3.1 Étapes

Pour **chaque opérateur** que vous ajoutez, vous **devez effectuer** les 6 étapes décrites ci-dessous.

1. Ajoutez le corps de la fonction réalisant cet opérateur dans le fichier `tai.c`. Vous pouvez éventuellement la faire précéder de fonctions auxiliaires ; dans ce cas, vous n'oublierez pas d'utiliser le mot-clé `static`.
2. Ajoutez l'en-tête de cette fonction dans le fichier `tai.h`.
3. Créez le fichier source permettant d'appeler la fonction précédente dans une fonction `main` afin de produire un programme exécutable. Vous suivrez les modèles donnés par `inversion.c` et `hist2im.c`.
4. Ajoutez tout ce qui est nécessaire dans le fichier `Makefile` pour pouvoir produire le nouvel exécutable (des commentaires à l'intérieur du fichier sont là pour vous guider).
5. Lancez la compilation avec la commande `make`.
6. Mettez au point et testez votre opérateur. Pour afficher les images au formats PBM, PGM et PPM, vous pouvez utiliser la commande `display` d'ImageMagick. Pour afficher les matrices, vous pouvez utiliser un éditeur de texte ou, plus simplement, les commandes `cat` ou `more`.

3.2 Mise en conformité

Pour chaque opérateur que vous ajoutez, assurez-vous qu'il **respecte les directives** de l'énoncé et qu'il **suiv le modèle** donné par les opérateurs `inversion` et `hist2im`. En particulier, il est **obligatoire** de :

1. respecter la syntaxe d'appel décrite dans l'énoncé : nom de l'exécutable (attention aux minuscules), ordre des paramètres ;
2. effectuer les vérifications sur les paramètres passés au programme : afficher la syntaxe de l'opérateur sur `stderr`, sortir avec le bon code de retour ;
3. afficher l'aide si l'opérateur est appelé avec l'option `-h` ;
4. rendre robuste l'opérateur en prévoyant, par exemple, que l'utilisateur peut représenter un vecteur par une matrice contenant une seule ligne ou une seule colonne (LUT, histogrammes).

3.3 Liste des opérateurs à programmer

Ajoutez, l'un après l'autre les opérateurs décrits ci-dessous.

1. `applut ImageEntree.pgm lut.mat ImageSortie.pgm`
   Cet opérateur applique une transformation ponctuelle à une image. La transformation est donnée au programme sous la forme d'une matrice de `int` (LUT) stockée dans un fichier au format `Matrix`. Vous pouvez tester ce programme avec le fichier `lut-inv.mat`.
2. `hist ImageEntree.pgm Histogramme.mat`
   Cet opérateur calcule l'histogramme d'une image de niveaux de gris sous la forme d'une matrice de taille 1×256 stockée dans un fichier au format `Matrix`. Vous pouvez vérifier que vous obtenez le même histogramme pour `lena.pgm` que celui qui se trouve dans `histlena.mat`.
3. `histeg ImageEntree.pgm Histogramme.mat`
   Cet opérateur calcule l'histogramme égalisé désiré associé à une image. Seul le nombre de pixels de l'image est utile pour calculer cet histogramme (le nombre de niveaux de gris étant égal à 256).
4. `hist2cumhist Histogramme.mat HistogrammeCumule.mat`
   Cet opérateur calcule l'histogramme cumulé correspondant à un histogramme.
5. `specifhist HistogrammeCumule.mat HistogrammeCumuleDesire.mat lut.mat`
   Cet opérateur calcule la transformation ponctuelle (`lut.mat`) donnée par l'algorithme de spécification d'histogramme à partir de l'histogramme cumulé d'une image et d'un histogramme cumulé désiré.

Testez les opérateurs précédents en réalisant une égalisation d'histogramme de l'image `lena.pgm`.

6. `mat2im [-p|-r] Matrice.mat Image.pgm`
   Cet opérateur transforme une matrice de `double` en une image de niveaux de gris. La stratégie utilisée est spécifiée par l'option. L'option `-p` indique de choisir, pour chaque pixel, l'entier le plus proche appartenant à l'intervalle $[0, 255]$. C'est l'option par défaut. L'option `-r` indique d'effectuer un « recadrage affine de la dynamique » vers l'intervalle $[0, 255]$ suivi d'un arrondi à l'entier le plus proche.
7. `conv [-c|-n|-b] Image.pgm Masque.mat MatriceResultat.mat`
   Cet opérateur réalise la convolution d'une image par un masque stocké sous la forme d'une matrice de `double` dans un fichier au format `Matrix`, le résultat étant stocké sous la forme d'une matrice de `double` dans un fichier au format `Matrix`. La stratégie de gestion des bords de l'image est spécifiée par l'option. Avec l'option `-c`, les valeurs initiales des pixels du bord sont copiées dans la matrice résultat. C'est l'option par défaut. Avec l'option `-n`, les pixels du bord sont mis à zéro. Avec l'option `-b`, les pixels du bord sont mis à 255. L'épaisseur des bords dépend de la taille du masque de convolution. On suppose que le masque est rectangulaire, avec un nombre de lignes et un nombre de colonnes impairs, et que son origine est son centre.

Pour tester les deux opérateurs précédents, utilisez le fichier `moy3x3.mat` qui contient le masque 3×3 du filtre moyenne permettant de lisser une image de niveaux de gris. Utilisez aussi `sobelh.mat` et `sobelv.mat` qui contiennent les deux masques de Sobel permettant de calculer les deux composantes du vecteur gradient de chaque pixel de l'image. Vous pouvez également créer vos propres masques de convolution avec un simple éditeur de texte (le format `Matrix` n'est pas un format binaire).

8. `bin ImageEntree.pgm Seuil ImageSortie.pbm`
   Cet opérateur « binarise » une image en niveau de gris en appliquant un seuil global. Pour chaque pixel, sa valeur dans l'image résultat est 0 si son niveau de gris est strictement inférieur au seuil, 1 sinon. L'image en entrée doit être en niveaux de gris (PGM). Le seuil doit être un entier dans l'intervalle $[1, 254]$. L'image en sortie est binaire (PBM).
9. `otsu Histogramme.mat`
   À partir d'un histogramme, cet opérateur affiche sur la sortie standard le niveau de gris (et seulement ce nombre suivi du caractère `'\n'`) qui est le seuil optimal selon la méthode d'Otsu. Si vous appliquez cet opérateur sur l'histogramme de l'image `lena.pgm`, il doit afficher `115`.

Vous pouvez, par exemple, tester les deux opérateurs précédents sur l'image `enveloppe.pgm`.

10. `erode ImageEntree.pbm ElementStructurant.mat ImageSortie.pbm`
    Cet opérateur réalise l'érosion de l'image binaire `ImageEntree.pbm` par l'élément structurant (ES) `ElementStructurant.mat` et écrit le résultat dans `ImageSortie.pbm`. L'ES est constitué d'une matrice de `int` au format `Matrix` dont le nombre de lignes et le nombre de colonnes sont impairs et qui contient uniquement des 1 et des 0. Son origine est son centre. Attention, les pixels du bord de l'image doivent être traités en ne considérant que la partie de l'ES qui se trouve à l'intérieur de l'image. Cette manière de traiter les pixels du bord nécessite une gestion fine des indices des boucles. Vous pouvez tester cet opérateur en érodant l'image `im-bin.pbm` avec l'ES `carre3x3.mat`. Vous devez obtenir la même image que `res-ero.pbm`.
11. `dilate ImageEntree.pbm ElementStructurant.mat ImageSortie.pbm`
    Cet opérateur doit fonctionner comme le précédent, mais il réalise une dilatation. Vous pouvez tester cet opérateur en dilatant l'image `im-bin.pbm` avec l'ES `carre3x3.mat`. Vous devez obtenir la même image que `res-dil.pbm`.

Pour tester les deux opérateurs précédents, vous pouvez réaliser l'ouverture de l'image `disques-et-tiges.pbm` avec l'ES `disque11x11.mat`. Pour tester les quatre opérateurs précédents, vous pouvez créer une chaîne d'opérateurs permettant de détecter les cellules présentes dans l'image `cel.pgm` :

- calcul du seuil de binarisation avec la méthode d'Otsu ;
- binarisation de l'image ;
- inversion de l'image ;
- ouverture de l'image (une érosion suivie d'une dilatation) avec le disque `disque7x7.mat` comme ES.

Vous pouvez aussi observer l'évolution du résultat si vous remplacez `disque7x7.mat` par `disque11x11.mat`.

3.4 Évaluation

En plus du travail effectué durant chaque séance, vous devez :

- créer une archive au format ZIP dont le nom suit le modèle `TAI-NOM-Prenom.zip` dans lequel vous remplacerez `NOM` par votre nom et `Prenom` par votre prénom, sans utiliser de lettres accentuées ; cette archive ne doit pas contenir de sous-répertoire (tous les fichiers doivent être au même niveau) ; elle doit contenir les fichiers suivants : `Makefile`, `limace.h`, `limace.c`, `tai.h`, `tai.c` et les fichiers contenant les fonctions `main` de chaque opérateur ; cette archive ne doit contenir ni les fichiers exécutables, ni les fichiers objets (`.o`), ni les fichiers de sauvegarde (`.bak`, `~`, etc.) de l'éditeur que vous utilisez, ni les fichiers qui vous ont été donnés pour tester vos opérateurs (`lena.pgm`, `cel.pgm`, `lut-inv.mat`, etc.) ; les fichiers ne doivent contenir aucun caractère accentué ; si vous disposez de la commande `zip`, l'archive peut être fabriquée en tapant :
  ```
  zip TAI-NOM-Prenom.zip Makefile *.h *.c
  ```
- déposer cette archive sur Moodle à une date qui vous sera donnée ultérieurement : dès que l'espace de dépôt sera ouvert, l'information sera donnée sur le wiki enseignement.

Cette archive va être utilisée pour :

- vérifier le bon fonctionnement de vos programmes (ils vont être un peu « torturés ») ;
- évaluer la qualité du code que vous avez écrit (pertinence des algorithmes, clarté, indentation, commentaires, respect des consignes, etc.) ;
- détecter d'éventuelles copies ou plagiats.

La note de contrôle continu pour la partie concernant le traitement et l'analyse d'images sera obtenue à partir du travail effectué durant les séances (sur 15) et des programmes fournis à la fin (sur 5).