

TP n° 1

Prise en main de Pandore

Opérations élémentaires sur les images

But du TP

- Prise en main de Pandore.
- Manipulation d'opérateurs élémentaires.
- Construction d'applications simples (plans d'opérateurs) de traitement d'images : génération, visualisation, seuillage, combinaison linéaire, transformations géométriques.

Remarques générales sur les TPs

- Il est impératif de créer un répertoire par TP.
- Dans la mesure du possible, écrivez des scripts Shell lors de l'enchaînement des opérateurs.
- On vous demandera périodiquement des rapports de TPs.
- Vous serez notés sur le travail individuel dans le binôme, sur les réponses apportées aux questions, sur la qualité de vos solutions et de vos rapports.

1 Introduction à Pandore

1.1 Qu'est ce que Pandore ?

Pandore est une bibliothèque standardisée d'opérateurs de traitement d'images. Elle est développée au sein de l'équipe Image du laboratoire GREYC (UMR CNRS 6072). La version actuelle regroupe des opérateurs traitant d'images 1D, 2D et 3D, en niveaux de gris, en couleurs et multispectrales. Elle se compose :

1. d'une collection d'opérateurs exécutables.
2. d'un environnement de programmation en C++.

1.2 A qui s'adresse Pandore ?

Pandore s'adresse en priorité aux traiteurs d'images, dans la mesure où elle nécessite des connaissances sur les opérations de traitement d'images et sur la façon de les combiner pour résoudre un problème entier.

L'utilisation de la bibliothèque n'exige aucune compétence particulière en programmation puisque les opérateurs s'emploient comme des commandes exécutables du système d'exploitation. Par contre, le développement de nouveaux opérateurs nécessite de bonnes connaissances en programmation C++.

1.3 Comment fonctionne Pandore ?

Cette bibliothèque se conçoit comme une collection de programmes exécutables (opérateurs) travaillant directement sur des fichiers image. La construction d'une application de traitement d'images se fait par l'activation de succession d'opérateurs, les images de sortie des uns servant d'images d'entrée aux autres.

Remarque importante Les images manipulées sous Pandore sont forcément au format Pandore, mais il existe des opérateurs de conversion des formats standards (e.g., bmp, gif, tiff, etc).

1.4 Documentation et plus d'infos

Pour plus d'informations, aussi bien l'utilisateur que le développeur sont invités à consulter la site web dédié à Pandore à l'adresse : <https://clouard.users.greyc.fr/Pandore/>. Une documentation détaillée peut y être consultée aussi bien sur l'utilisation des opérateurs atomiques et moléculaires que sur le développements de nouveaux opérateurs. Par ailleurs, un manuel en ligne est aussi disponible par l'opérateur `pman` sous Pandore.

2 Prise en main de Pandore

2.1 Lancement de Pandore

La bibliothèque Pandore a été installée dans le répertoire : `/opt/pandore`. Pour devenir un utilisateur de Pandore, il suffit d'exécuter le script de configuration :

```
/opt/pandore/pandore
```

A partir de là, toutes les commandes doivent être accessibles. Pour cela essayer l'opérateur `pversion` par la commande Unix (le caractère `$` symbolise le prompt Unix) :

```
$ pversion
```

Il est possible de définir de manière durable, l'accès à Pandore. Pour cela, il suffit de redéfinir la variable d'environnement : `PATH` en y ajoutant le chemin `/opt/pandore/`.

2.2 L'utilisation des opérateurs

Tous les opérateurs de la bibliothèque sont des commandes exécutables du système d'exploitation, c'est-à-dire invocables simplement à partir de la ligne de commande du système d'exploitation. Il est ainsi possible d'écrire très simplement des scripts enchaînant les appels à ces programmes.

Les opérateurs de la bibliothèque respectent tous la même syntaxe d'appel :

```
$ operateur parametre* [-m im_msq] [im_src|-]* [im_dest|-]*
```

où seule la présence des paramètres, en nombre et en type, est obligatoire. Si le masque `im_msq` est absent l'opérateur agit sur toute l'image, sinon il n'agit sur les parties indiquées par le masque `im_msq`. Un masque est un fichier image au format Pandore.

Exercice 1

Appliquer une rotation de 45° sur une l'image `/home/public/pandore-images/lena.bmp`, puis visualiser le résultat avec la séquence de commandes suivante :

```
$ pbmp2pan /home/public/pandore-images/lena.bmp lena.pan
$ protation 0 45 lena.pan lena_rot.pan
$ pvisu lena_rot.pan
```

Les autres arguments peuvent être omis : Si les noms des images *im_src* et *im_dest* sont absents de la ligne de commande, ou s'ils sont remplacés par le signe "-", c'est à partir de l'entrée (resp. vers la sortie) standard que sont lues (resp. écrites) les données. L'exemple ci-dessus peut être réécrit :

```
$ pbmp2pan /home/public/pandore-images/lena.bmp | protation 0 45 - - | pvisu
```

De plus, les opérateurs Pandore jouissent d'une généralité dans le sens où un même opérateur peut agir sur plusieurs types de fichiers Pandore (e.g. images en niveaux de gris, image couleurs, carte de régions, etc.). Par ailleurs, chaque opérateur Pandore retourne un résultat qui pourra être utilisé par la suite comme paramètre d'un autre opérateur ou comme variable de contrôle. Le résultat de l'exécution d'un opérateur est soit :

- La chaîne de caractères **FAILURE** pour une sortie en erreur.
- La chaîne de caractères **SUCCESS** pour une succès sans valeur de sortie.
- Une valeur numérique de n'importe quel type de base (char, ... , float, double). La récupération de cette valeur se fait par la commande **pstatus** sous Pandore 6 après l'exécution de l'opérateur à interroger.

Exercice 2

1. Afficher la valeur retournée par les opérateurs *pbmp2pan* et *protation*.
2. Afficher le maximum d'une image en utilisant l'opérateur *pmaximumvalue*.

3 Opérations élémentaires sur les images

3.1 Visualisation, analyse

- Convertir l'image `/home/public/pandore-images/lena.bmp` au format Pandore.
- Afficher les propriétés de cette image (*pfile*).
- Visualiser l'image résultante.
- Tester les différentes possibilités offertes par le visualiseur : valeurs, histogramme, profil ligne et colonne, lut, zoom, etc.

3.2 Images et espaces couleurs

L'image `/home/public/pandore-images/rondelles.bmp` est une image couleur dans l'espace RVB (Rouge-Vert-Bleu).

- Extraire et visualiser chacune des trois composantes de l'image RVB à l'aide de l'opérateur *pimc2img*.
- A l'aide de la commande *pstatus*, afficher la valeur retournée par les opérateurs *pbmp2pan* et *pimc2img*.
- Afficher le maximum de chacune des composantes Rouge, Verte et Bleue en utilisant l'opérateur *pmaximumvalue*.
- Convertir l'image couleurs de l'espace RVB à l'espace HSL (Hue-Saturation-Luminance), à l'aide de l'opérateur *prgb2hsl*. Visualiser l'image obtenue. Qu'observez-vous ?

- Extraire et visualiser chacune des trois composantes de l'image HSL toujours à l'aide de l'opérateur `pimc2img`. Comparer aux composantes RVB.
- Répéter les mêmes opérations avec d'autres espaces couleurs vus en cours.

3.3 Images synthétiques

- Tester les différentes possibilités de l'opérateur `pshapedesign` en créant au moins un exemplaire de chaque forme (taille des images 256×256) : disque, carré, rectangle.

3.4 Bruit

- Sur l'une des images synthétiques précédentes, générer différents types de bruit offerts par l'opérateur `paddnoise`.
- Observer pour chaque bruit, d'histogramme avant et après contamination. Que remarquez-vous ? Interprétez.

3.5 Transformations géométriques

Opérateurs utiles : `pshapedesign`, `paddnoise`, `ptranslation`, `protation`, `presizing`, `pbinarization`.

- Générer une image de rectangle. non centré et non aligné sur les bords de l'image.
- Corrompre cette image par un bruit additif Gaussien avec un $RSB=20dB$.
- Appliquer à l'image bruitée une translation pour la centrer.
- Lui appliquer une rotation pour aligner le rectangle sur les bords de l'image.
- Re-dimensionner l'image pour transformer le rectangle en carré.
- A partir de l'histogramme, proposer un seuil pour binariser l'image et obtenir un masque sur le carré.

3.6 Amélioration d'images

Opérateurs utiles : `phistogramequalization`, `plineartransform`, `psharp`.

- Convertir l'image `/home/public/pandore-images/aquitain.bmp` au format Pandore. Que remarquez-vous ?
- Réhaussez son contraste par modification de son histogramme. Qu'observez-vous ? Interprétez les résultats obtenus par `phistogramequalization` et par `plineartransform`.
- Réhaussez le contraste par mesure différentielle du 2nd ordre (opérateur `psharp`). Qu'observez-vous ?