

# TP : Outils

Jonathan Fabrizio  
<http://jo.fabrizio.free.fr/>

## Objectif

L'objectif de ce T.P. est de survoler des outils que vous pouvez utiliser en *batch* ou pour prototyper. Ce n'est évidemment pas exhaustifs. A vous de trouver les outils que vous trouvez *vous* ergonomique. Les solutions proposées ne seront peut-être pas celles que vous retiendrez...

## 1 Les formats de fichiers

Il y a quantité de format d'images différents, il faut donc être capable d'identifier correctement le format d'une image et de pouvoir convertir une image d'un format à un autre aisément.

### 1.1 Convertir d'un format à un autre

#### Convertir une image

L'image du répertoire q111 est au format JPG. Convertissez la au format TGA puis PNG et enfin PPM. Comparez les tailles et les qualités. Regardez le contenu de l'image PPM (ou au moins le debut). A quoi correspondent les valeurs.

#### Convertir un lot d'images

Regardez maintenant le lot d'images du répertoire q112. Elles sont au format PNG. Convertissez les toutes au format TIFF.

### 1.2 Identifier un format d'image

#### Un problème sur une image

Regardez l'image du répertoire q121 et exécutez votre tp1 sur cette image. Cela ne fonctionne pas ! Pourquoi ? Essayez d'identifier le problème.

#### Un problème sur une autre image

Regardez l'image du répertoire q122 et exécutez votre tp1 sur cette image. Cela ne fonctionne pas non plus ! Pourquoi ? Essayez d'identifier le problème.

#### EXIF

Regardez l'image du répertoire q123 et trouvez le modèle de couleur utilisé<sup>1</sup>. Regardez les autres informations disponibles dans le EXIF. Essayez sur d'autres images.

### 1.3 Modifier l'enregistrement d'une image

#### Encodage, compression, couleur...

Corriger les encodages des deux images précédentes pour pouvoir lancer le tp1 dessus. Ajoutez ou enlevez la compression de l'encodage d'une image.

---

1. Attention, cette information n'est pas toujours vraiment fiable

## 2 Actions sur un lot d'images

### *Cropper un lot d'images*

Sur le lot d'images du répertoire `q112`, *cropper* une vignette de  $10 \times 10$  pour chaque image. Essayez aussi de changer la taille des images, avec différentes interpolations.

## 3 C'est le moment de faire un pause

Pour cela, lisez la doc d'imagemagick <https://imagemagick.org/script/command-line-tools.php> et particulièrement la doc des outils `identify`, `convert` et `mogrify`. Une fois la pause finie, repassez les sections précédentes en revu et complétez vos solutions avec l'aide de ces outils.

## 4 Gimp

### 4.1 Interface

#### Prise en main de l'interface

Prenez le temps d'essayer l'interface, de voir les filtres (**Filtres->**) et les outils (dans **Couleurs->**) disponibles. Ensuite, vérifiez les résultats de votre TP1 :

- égalisation d'histogramme (niveaux de gris)
- égalisation d'histogramme (couleur)
- amélioration de la netteté avec le Laplacien

Prenez le temps d'inspecter les différents espaces de couleurs.

### 4.2 Gimp en ligne de commande

#### Appel de filtres

Invoquer gimp en ligne de commande pour appliquer un filtre gaussien selectif. Inspirez-vous de l'exemple suivant (oui c'est du *scheme*, voir la doc de gimp) :

Contenu du fichier `~/gimp-x.x/scripts/unsharp.scm` :

```
(define (unsharp-image filename filename-out)
  (let* ((image (car (gimp-file-load RUN-NONINTERACTIVE filename filename)))
        (drawable (car (gimp-image-get-active-layer image))))
    (plug-in-unsharp-mask RUN-NONINTERACTIVE
      image drawable 5 0.5 0)
    (gimp-file-save RUN-NONINTERACTIVE image drawable filename-out filename-out)
    (gimp-image-delete image)))
```

puis appel de la fonction depuis la ligne de commande :

```
gimp -i -b '(unsharp-image "nuage.tiff" "nuage-out.tiff")' -b '(gimp-quit 0)'
```

Ecrivez votre fonction dans un fichier `gauss_sel.scm` (dans le répertoire `~/gimp-x.x/scripts/`). Améliorez la fonction en passant les arguments de la fonction (arguments de la gaussienne) depuis la ligne de commande.

#### Votre propre filtres (Optionnel)

Ecrivez un script qui réalise l'amélioration de la netteté à l'aide du Laplacien. Pour cela vous aurez besoin de `plug-in-convmatrix` :

```
(plug-in-convmatrix 1 image drawable nombre_d_éléments_dans_le_noyau
  cons-array_noyau_de_convolution TRUE diviseur decalage
  taille_array_channels cons-array_channels 0)
cons-array_channels indique les canaux à traiter (1) ou pas (0). Il contient à
l'index 0 : gray, 1 : red, 2 : green, 3 : blue, 4 : alpha.
```

Pour déclarer un `cons-array` :

```
(let* (channels (cons-array 5 'long) )
  (matrix (cons-array 25 'double)))
```

Pour écrire dans un cons-array :

```
(aset channels 0 1)
(aset channels 1 1)
(aset channels 2 1)
(aset channels 3 1)
(aset channels 4 0)
```

Vous pouvez lire aussi la doc <https://docs.gimp.org/2.10/fr/gimp-using-script-fu-tutorial.html>.

### **Ajout de votre filtre dans l'interface (Optionnel)**

Faites en sorte que votre script soit visible depuis l'interface graphique de Gimp. Pour cela vous devez utiliser `script-fu-register`