

TP (Solutions) : Outils

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

1 Les formats de fichiers

1.1 Convertir d'un format à un autre

Convertir une image

```
convert nuage.jpg nuage.tga
convert nuage.jpg nuage.png
convert nuage.jpg nuage.ppm
head -n 4 nuage.ppm
head -n 5 nuage.ppm
```

Convertir un lot d'images

```
mogrify -format tiff *.png
```

1.2 Identifier un format d'image

Un problème sur une image

```
identify -verbose nuage01.tga
```

C'est un TGA avec un canal alpha.

Un problème sur une autre image

```
identify nuage02.tga
```

Le fichier a l'extension TGA mais ce n'est pas du TGA.

EXIF

```
identify -verbose DSC00663.JPG
```

1.3 Modifier l'enregistrement d'une image

Encodage, compression, couleur...

```
convert nuage01.tga -alpha off nuage01_ok.tga
convert nuage02.tga -alpha off nuage02_ok.tga
convert nuage02_ok.tga -compress none nuage_no.tiff
convert nuage02_ok.tga -compress jpeg nuage_jpeg.tiff
```

2 Actions sur un lot d'images

Cropper un lot d'images

```
for file in *.png
do
    convert ${file} -crop 10x10+0+0 +repage ${file%.png}_c.png
done
```

3 xxx

4 Gimp

4.1 Interface

Prise en main de l'interface

En faisant attention au canal sélectionné :

Couleurs -> auto -> égaliser

Pour voir l'histogramme :

Fenêtres -> fenêtres ancrables -> histogramme

A voir aussi :

Couleurs -> courbes

Pour le Laplacien :

Filtres-> génériques -> matrice de convolution

Pour les différents canaux :

Couleurs -> composants -> décomposer (par images)

4.2 Gimp en ligne de commande

Appel de filtres

```
(define (gauss-sel filename filename-out radius maxdelta)
  (let* ((image (car (gimp-file-load RUN-NONINTERACTIVE filename filename)))
        (drawable (car (gimp-image-get-active-layer image))))
    (plug-in-sel-gauss RUN-NONINTERACTIVE
      image drawable radius maxdelta)
    (gimp-file-save RUN-NONINTERACTIVE image drawable filename-out filename-out)
    (gimp-image-delete image)))
```

Puis :

```
gimp -i -b '(gauss-sel "nuage.tiff" "nuage-sel.tiff" 10 10)' -b '(gimp-quit 0)'
```

Votre propre filtres

```
(define (sharpen-laplace filename filename-out)
  (define (get-matrix matrix-list)
    (let* ((count 0)
          (argc_matrix 25)
          (matrix (cons-array argc_matrix 'double)))
      (while (< count argc_matrix)
        (aset matrix count (nth count matrix-list))
        (set! count (+ count 1)))
      matrix))

  (let* ((image (car (gimp-file-load RUN-NONINTERACTIVE filename filename)))
        (drawable (car (gimp-image-get-active-layer image)))
        (channels (cons-array 5 'long) ))
    (aset channels 0 1)
    (aset channels 1 1)
    (aset channels 2 1)
    (aset channels 3 1)
    (aset channels 4 0)
    (plug-in-convmatrix 1 image drawable 25
```

```
(get-matrix
      (list 0 0 0 0 0
            0 0 -1 0 0
            0 -1 5 -1 0
            0 0 -1 0 0
            0 0 0 0 0)) TRUE 1 0 5 channels 0)
(gimp-file-save RUN-NONINTERACTIVE image drawable filename-out filename-out)
(gimp-image-delete image))

gimp -i -b '(unsharp-laplace "nuage.tiff" "nuage-lap.tiff")' -b '(gimp-quit 0)'
```

Ajout de votre filtre dans l'interface

Faites en sorte que votre script soit visible depuis l'interface.

```
(define (sharpen-laplace image drawable)
  (define (get-matrix matrix-list)
    (let* ((count 0)
           (argc_matrix 25)
           (matrix (cons-array argc_matrix 'double)))
      (while (< count argc_matrix)
        (aset matrix count (nth count matrix-list))
        (set! count (+ count 1)))
      matrix))

  (let* ((channels (cons-array 5 'long) )
         (aset channels 0 1)
         (aset channels 1 1)
         (aset channels 2 1)
         (aset channels 3 1)
         (aset channels 4 0)
         (plug-in-convmatrix 1 image drawable 25
                              (get-matrix
                                (list 0 0 0 0 0
                                      0 0 -1 0 0
                                      0 -1 5 -1 0
                                      0 0 -1 0 0
                                      0 0 0 0 0)) TRUE 1 0 5 channels 0)
         (gimp-displays-flush)))

  (script-fu-register
    "sharpen-laplace" ;func name
    "sharpen-laplace" ;menu label
    "sharpen-laplace." ;description
    "Jonathan Fabrizio" ;author
    "copyright 2018, Jonathan Fabrizio" ;copyright notice
    "March 20, 2018" ;date created
    "" ;image type that the script works on
    SF-IMAGE "SF-IMAGE" 0
    SF-DRAWABLE "SF-DRAWABLE" 0
  )

  (script-fu-menu-register "sharpen-laplace" "<Image>/Filters/Sharpen")
```