

TP - Introduction à la programmation sous ImageJ

Nicholas Journet - Traitement d'images - IUT - ¹

3.1 Programmation sous ImageJ

Récupérez l'archive *projetVideij.zip*. C'est un projet vide permettant d'utiliser les fonctionnalités d'ImageJ via un fichier *jar*.

Nous allons créer un premier code qui permet d'ouvrir une image en mémoire, d'accéder à la valeur de ses niveaux de gris, les modifier et affiche le résultat à l'écran.

Question 1

Créez le fichier *MonScript.java* en recopiant le code suivant :

```
1 public static void main(String[] args) {
2     //lance l'interface d'imageJ
3     new ImageJ();
4     //Chargement d'une image dans un objet
5     ImagePlus image = IJ.openImage("path-to-image-test");
6     //affichage d'une popup
7     IJ.showMessage("Accès à ImageJ depuis le code source");
8     //conversion de l'image en NdG
9     new ImageConverter(image).convertToGray8();
10    //Récupération des NdG
11    ImageProcessor ip = image.getProcessor();
12    byte[] pixels = (byte[]) ip.getPixels();
13    //l'image est stockée dans un tableau 1D de byte.
14    int height = ip.getHeight();
15    int width = ip.getWidth();
16    //parcours du tableau 1D
17    for (int i = 0; i < height; i++) {
18        for (int j = 0; j < width; j++) {
19            //cast byte --> int
20            int pix = pixels[i * width + j] & 0xff;
21            //binarisation de l'image
22            if (pix < 120) {
23                pixels[i * width + j] = (byte) 0;
24            } else {
25                pixels[i * width + j] = (byte) 255;
26            }
27        }
28    }
29    //affichage de l'image via imageJ
30    image.show();
31    WindowManager.addWindow(image.getWindow());
32 }
```

Le type `byte` en Java est signé et prend sa valeur entre -128 et 127, alors que nous voudrions un niveau de gris sur 8 bits entre 0 et 255. Si l'on caste un `byte` vers un autre type, il faut s'assurer que le bit de signe est éliminé. Ceci est fait très facilement avec un **et** logique

1. Support inspiré du tutoriel ImageJ de l'IJM.

```

1  int pix = pixels[i] & 0xff;    // conversion en int
2  ...
3  pixels[i] = (byte) pix;        // re-conversion en byte

```

Question 2

Créez la fonction `public static double meanImage(ImageProcessor ip)` qui permet de calculer la moyenne des niveaux de gris d'une image.

Question 3

Complétez le code suivant afin qu'il permette de connaître quelle image parmi celles d'un répertoire ressemble le plus à une image chopisie à l'avance. Le code permettant d'obtenir les fichiers présents dans un répertoire vous est donné (voir `listFiles`).

```

1
2
3
4  String path = " _____ ";
5  File[] files = _____ ;
6
7  if ( files != _____ ){
8      double gap = Double.MAX_VALUE;
9      _____
10     _____
11
12     for(int i=0; i < _____ ; i++ ){
13         if( ! files[i].isHidden() ){
14             // Création d'une image temporaire
15             String filePath = _____ .getAbsolutePath();
16             ImagePlus tempImg = new ImagePlus(filePath);
17
18             new ImageConverter( _____ ).convertToGray8();
19
20             ImageProcessor ipTemp = tempImg. _____ ;
21
22             // Calcul du niveau de gris moyen de l'image
23             double avgTemp = _____ ;
24
25             // Différence par rapport à l'image d'origine
26             double dif = Math.abs( _____ );
27
28             if ( dif < _____ ){
29                 _____
30             }
31         }
32     }
33
34     String closestImageName = _____ ;
35     IJ.showMessage("L'image la plus proche est " + closestImageName
36         + " avec une distance de " + gap + ".");
37 }
38
39
40 public File[] listFiles(String directoryPath){
41     File[] files = null;
42     File directoryToScan = new File(directoryPath);
43     files = directoryToScan.listFiles();
44     return files;
45 }

```

Question 4

Avec ImageJ, ajoutez du bruit à une image de la base fournie (*Process* → *Noise* → *Add Noise*). Le plugin de la question précédente *CompareImages* arrive t-il à retrouver l'image d'origine dans la base, et pourquoi ? Même question, mais cette fois-ci avec du bruit *Salt and Pepper*.

Question 5

Vous avez terminé ce petit tutoriel ? Maintenant vous pouvez commencer votre projet.



Ce document est publié sous Licence Creative Commons « By-NonCommercial-ShareAlike ». Cette licence vous autorise une utilisation libre de ce document pour un usage non commercial et à condition d'en conserver la paternité. Toute version modifiée de ce document doit être placée sous la même licence pour pouvoir être diffusée.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>