

ImageJ - TP 3

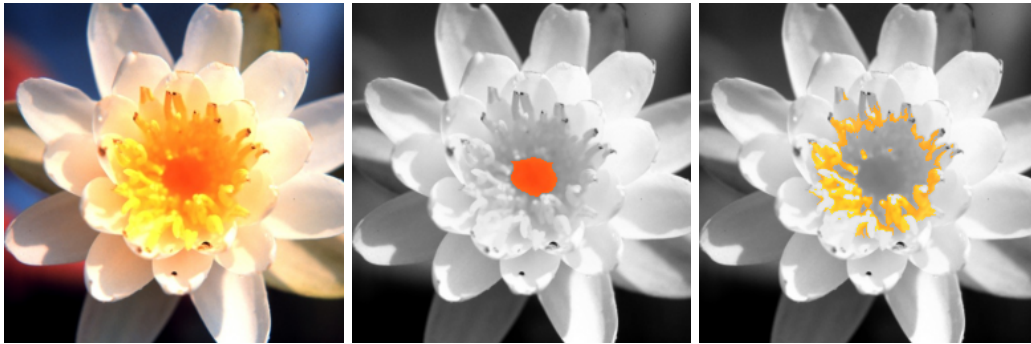
Un peu de couleurs !

Exercice 3.1 *Sélection de couleurs proches*

Le but de cet exercice est d'écrire un plugin de traitement d'image couleur qui :

- fait choisir une couleur de référence à l'utilisateur
- conserve dans l'image tous les pixels dont la couleur est proche de la couleur de référence
- grise tous les autres pixels

Par exemple, l'image initiale `fleur.png` (à gauche) a été transformée en choisissant la couleur rgb [250, 70, 20] (au centre) et la couleur [250, 200, 40] (à droite).



Vous complèterez le code fourni dans le fichier `Selection_Couleur.java`.

1. La méthode `couleurVersNg` renvoie la couleur grisée correspondant à la couleur rgb stockée dans l'entier donné en paramètre. On rappelle qu'il suffit de mettre le même niveau dans les trois canaux pour qu'une couleur apparaisse en gris. Le niveau de gris correspondant à une couleur $[r, g, b]$ peut être calculé par $0.3r + 0.59g + 0.11b$.
2. La méthode `distance` renvoie la distance entre deux couleurs $[r1, g1, b1]$ et $[r2, g2, b2]$. Cette distance est calculée par $(r1 - r2)^2 + (g1 - g2)^2 + (b1 - b2)^2$.
3. Dans la méthode `run`, une couleur de référence `couleurRef` est choisie par l'utilisateur et on fixe également une valeur de distance maximum `distanceMax`. Toutes les couleurs qui seront à une distance inférieure à `distanceMax` de la couleur de référence seront conservées, les autres seront grisées.
4. Retrouvez les résultats de l'exemple en traitant l'image `fleur.png`. Vous sauvez les résultats sous les noms `fleur-250-70-20.png` et `fleur-250-200-40.png`. Proposez deux autres exemples, avec des images de départ de votre choix.
5. Proposez un second plugin dont le rôle est le même que celui de la question 1 mais où la mesure de distance entre deux couleurs se fera sur la teinte. En d'autres termes, au lieu de mesurer leur distance dans le cube RGB, on va comparer la teinte des deux couleurs (première valeur du modèle de couleur HSB). Donnez des exemples de résultats.
Indication : le passage des valeurs RGB aux valeurs HSB peut se faire grâce à une méthode de la classe `Color`.
6. Pour plus de simplicité, il est possible de faire choisir une couleur directement sur l'image source. Pour cela, il suffit de faire cliquer l'utilisateur sur l'image et de récupérer la couleur aux coordonnées du clic par l'implémentation d'un `MouseListener`. Un exemple vous en est donné à l'adresse :
http://rsb.info.nih.gov/ij/plugins/download/Mouse_Listener.java.

- (a) Implémentez un plugin répondant à ce besoin qui crée une nouvelle image résultat à chaque clic sur l'image source.
- (b) Implémentez un second plugin qui ne créera qu'une image résultat et qui ajoutera les couleurs à l'image résultat une par une à chaque clic sur l'image source.

```
-----Selection_Couleur.java -----
import ij.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.PlugInFilter;
import ij.process.*;

public class Selection_Couleur implements PlugInFilter {

    int couleurVersNG(int couleur) {
        ...
    }

    double distance(int couleur1, int couleur2) {
        ...
    }

    public void run(ImageProcessor ip) {
        ColorChooser cc = new ColorChooser("couleur de référence", Color.yellow, true);
        Color cRef = cc.getColor();
        if (cRef == null) {
            IJ.error( "PlugIn cancelled" );
            return;
        }
        int couleurRef = (cRef.getRed() << 16) + (cRef.getGreen() << 8) + cRef.getBlue();
        int distanceMax = ...; // seuil à choisir

        ...
    }
}
```
