VISA – TP Segmentation

Arnaud Cojez

mercredi 16 novembre 2016

Table des matières

1 Introduction

Nous avons vu dans le précédent TP comment étaient représentées les couleurs en informatique. Dans ce TP, nous allons tenter de segmenter des images par classification de pixels.

Nous devrons écrire une Macro ImageJ qui permet de segmenter une image en un nombre donné de classes. Nous segmenterons ensuite d'autres images en utilisant ces classes et observerons les résultats.

2 Segmentation des images par K-Means et discrimination linéaire

2.1 Explication

Le but ici est d'écrire une macro qui permet dans un premier temps de segmenter une image en un nombre donné de classes. Pour ce faire nous utilisons l'algorithme supervisé K-Means grâce au plugin Segmentation > K-Means Clustering fourni.

Nous parcourons ensuite l'image Cluster Centroid Values afin de stocker les valeurs RGB des différentes classes.

Puis nous demandons à l'utilisateur d'ouvrir une nouvelle image.

Ensuite, nous segmentons la nouvelle image en sélectionnant pour chaque pixel la classe dont la couleur est la plus proche de la couleur courante (dans l'espace RGB).
La Macro a été fournie en pièce jointe.

2.2 Résultats

Nous testons dans un premier temps la macro sur l'image *cas_3_dalton15*, en demandant 6 classes. Nous obtenons l'image segmentée ci-dessous (à droite) ainsi que les valeurs des différentes classes.

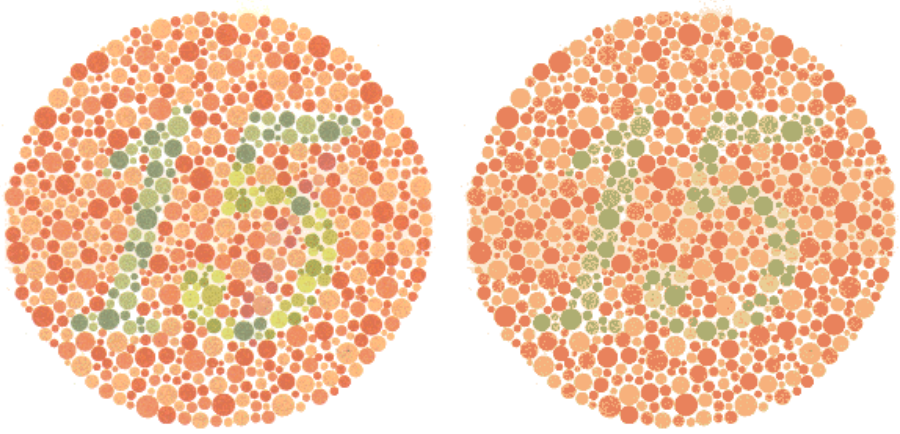

FIGURE 1 – À gauche : Image initiale | À droite : Image segmentée

1. R=255, G=255, B=255
2. R=252, G=231, B=207
3. R=247, G=177, B=124
4. R=234, G=206, B=148
5. R=233, G=130, B=092

6. R=174, G=174, B=114

Nous devons maintenant ouvrir une nouvelle image. Nous sélectionnons *cas_3_dalton74* pour la segmenter selon la méthode citée plus haut. Voici le résultat obtenu.

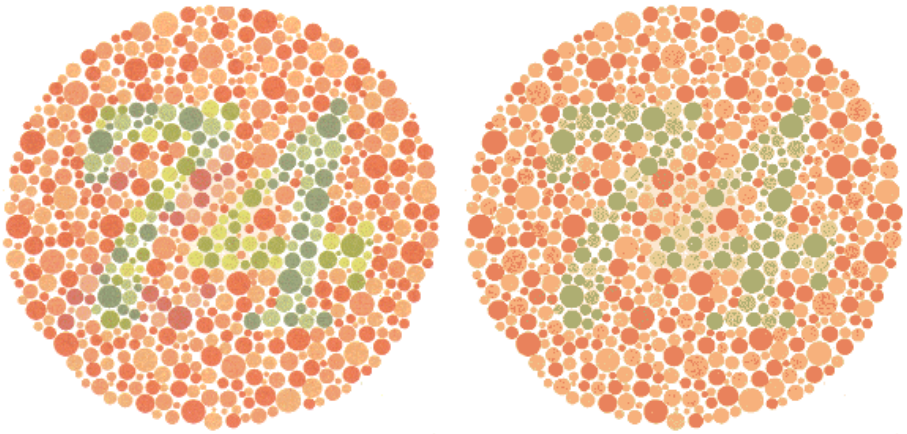

FIGURE 2 – À gauche : Image initiale | À droite : Image segmentée

3 Application aux images cas_4

3.1 Explication

Le but de cette partie est d'appliquer la macro codée dans la partie précédente et d'observer les résultats sur le lot d'images *cas_4*.

La première image selectionnée et segmentée est l'image *cas_4_dalton6*. Nous utiliserons les 6 classes extraites de cette image afin de segmenter les images suivantes.

3.2 Résultats

3.2.1 Image cas_4_dalton6

Après classification et segmentation par la méthod K-Means, nous obtenons l'image ci-dessous (à droite).

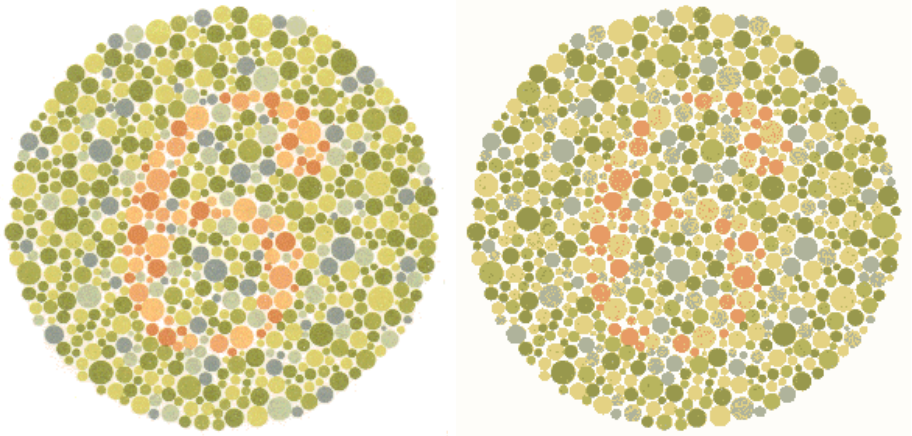

FIGURE 3 – À gauche : Image initiale | À droite : Image segmentée

Les classes obtenues sont les suivantes :

1. R=255, G=255, B=255
2. R=252, G=231, B=207
3. R=247, G=177, B=124
4. R=234, G=206, B=148
5. R=233, G=130, B=092
6. R=174, G=174, B=114

Nous observons l'image résultante dans les repères RGB et HSV, grâce au plugin Color Inspector 3D :

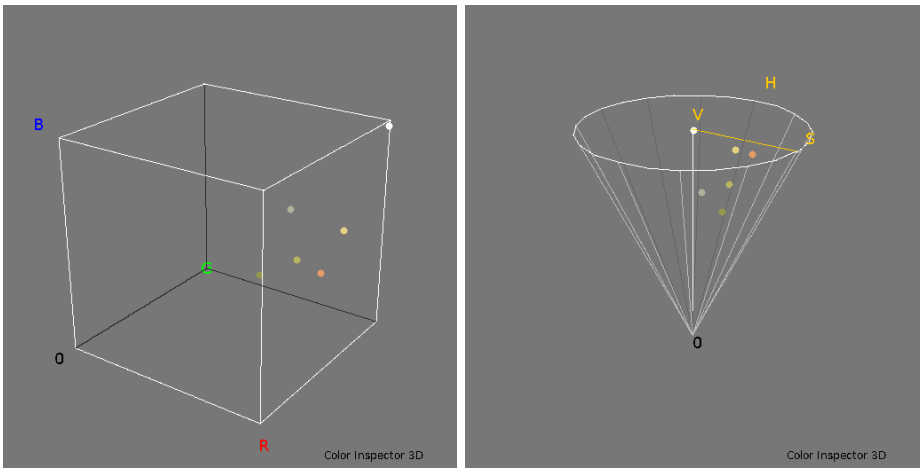

FIGURE 4 – À gauche : Vue dans l'espace RGB | À droite : Vue dans l'espace HSV

Nous ne voyons plus que 6 couleurs. Ces couleurs correspondent aux classes trouvées plus tôt. En effet, chaque pixel a pris la couleur de la classe la plus proche dans le repère RGB. D'où ces résultats.

3.2.2 Image cas_4_dalton8

Après segmentation, nous obtenons l'image suivante :

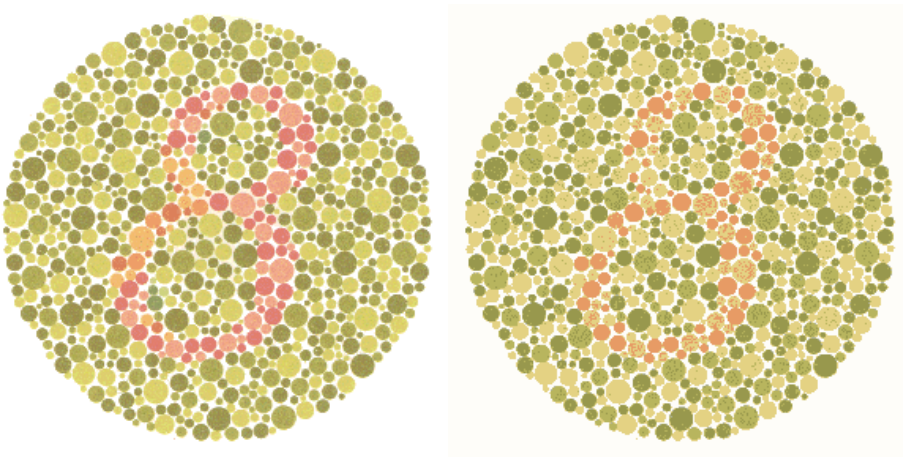

FIGURE 5 – À gauche : Image initiale | À droite : Image segmentée

Nous observons l'image résultante dans les repères RGB et HSV, grâce au plugin Color Inspector 3D :

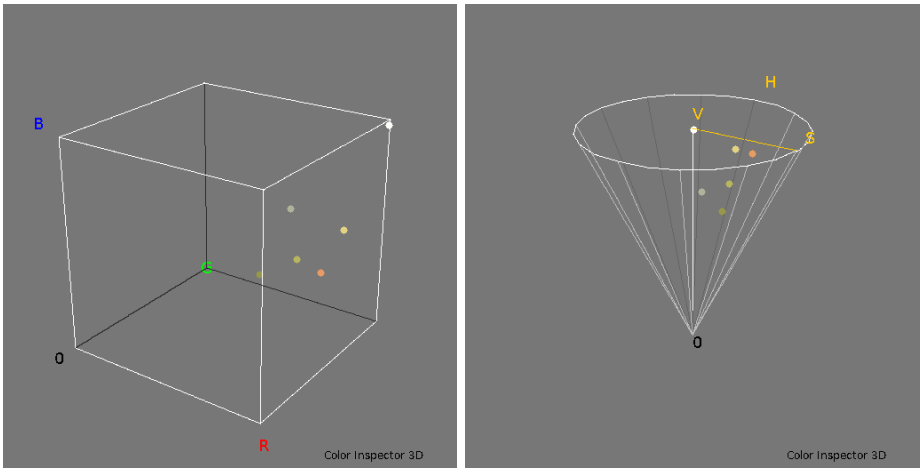

FIGURE 6 – À gauche : Vue dans l'espace RGB | À droite : Vue dans l'espace HSV

Ici également, nous voyons seulement 6 couleurs. Ces couleurs correspondent aux classes trouvées plus tôt.

3.2.3 Image cas_4_dalton29

Après segmentation, nous obtenons l'image suivante :

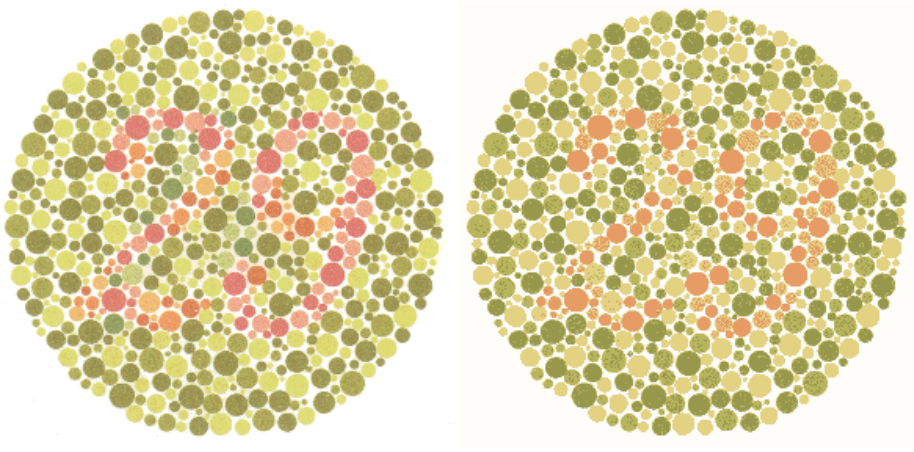

FIGURE 7 – À gauche : Image initiale | À droite : Image segmentée

Nous observons l'image résultante dans les repères RGB et HSV, grâce au plugin Color Inspector 3D :

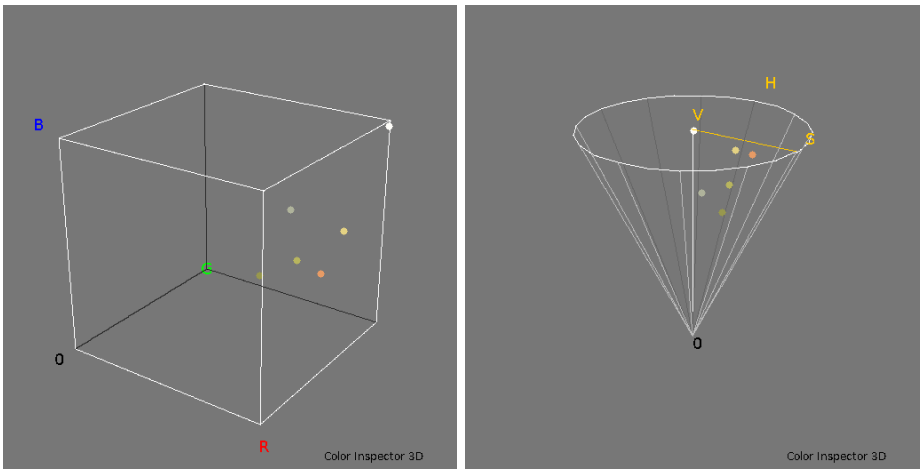

FIGURE 8 – À gauche : Vue dans l'espace RGB | À droite : Vue dans l'espace HSV

Ici également, nous voyons seulement 6 couleurs. Ces couleurs correspondent aux classes trouvées plus tôt.

3.2.4 Image cas_4_dalton45

Après segmentation, nous obtenons l'image suivante :

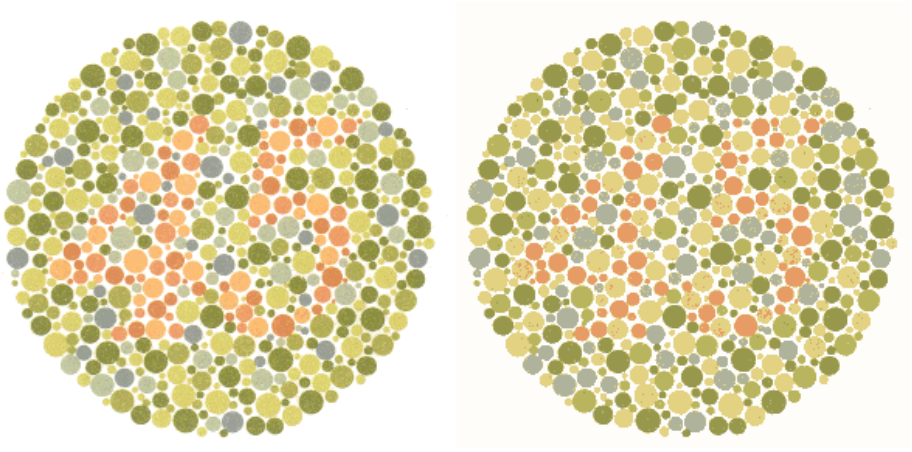

FIGURE 9 – À gauche : Image initiale | À droite : Image segmentée

Nous observons l'image résultante dans les repères RGB et HSV, grâce au plugin Color Inspector 3D :

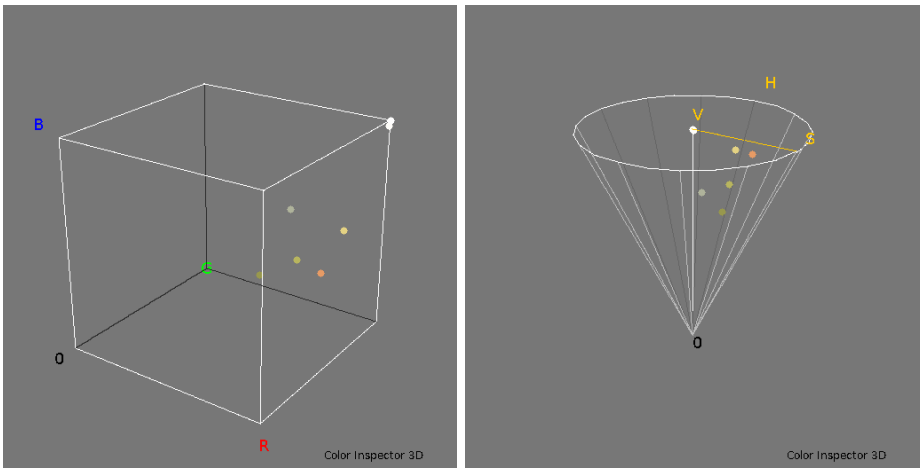

FIGURE 10 – À gauche : Vue dans l'espace RGB | À droite : Vue dans l'espace HSV

Ici également, nous voyons seulement 6 couleurs. Ces couleurs correspondent aux classes trouvées plus tôt.

3.2.5 Image cas_4_dalton_ligne

Après segmentation, nous obtenons l'image suivante :

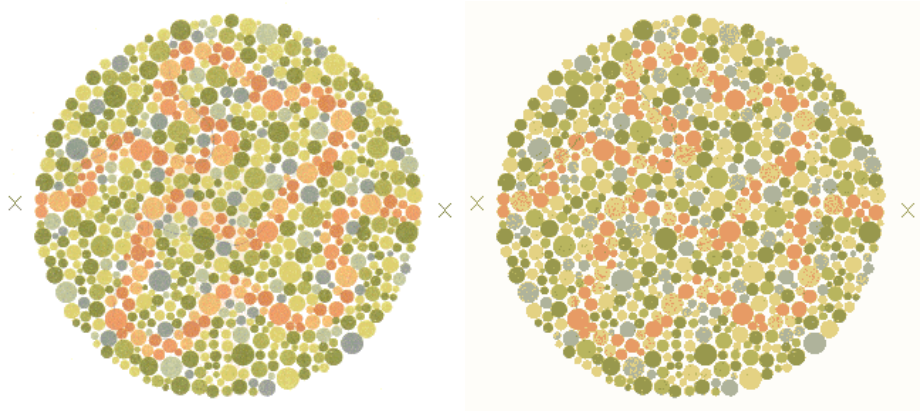

FIGURE 11 – À gauche : Image initiale | À droite : Image segmentée

Nous observons l'image résultante dans les repères RGB et HSV, grâce au plugin Color Inspector 3D :

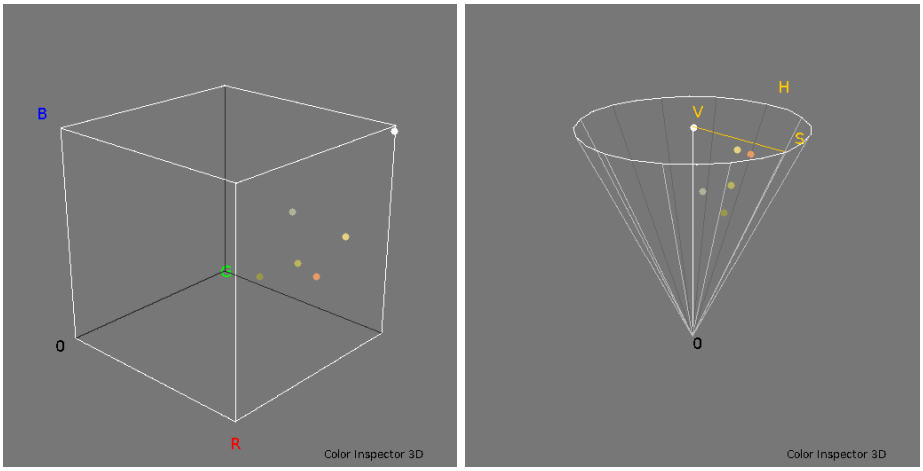

FIGURE 12 – À gauche : Vue dans l'espace RGB | À droite : Vue dans l'espace HSV

Ici également, nous voyons seulement 6 couleurs. Ces couleurs correspondent aux classes trouvées plus tôt.

4 Conclusion

Après avoir défini ce qu'étaient les couleurs et leurs représentations, nous avons pu segmenter des images en utilisant l'algorithme K-Means Clustering.

Cet algorithme nous a permis – une fois spécifié le nombre de classes voulues – d'obtenir les couleurs de chaque classes. Une fois ces couleurs stockées, nous avons pu segmenter des images similaires en récupérant pour chaque pixel, la classe la plus proche de sa couleur.

Nous avons observé que dans les espaces RGB et HSV, il n'y avait plus que les couleurs associées aux classes utilisées pour la segmentation.