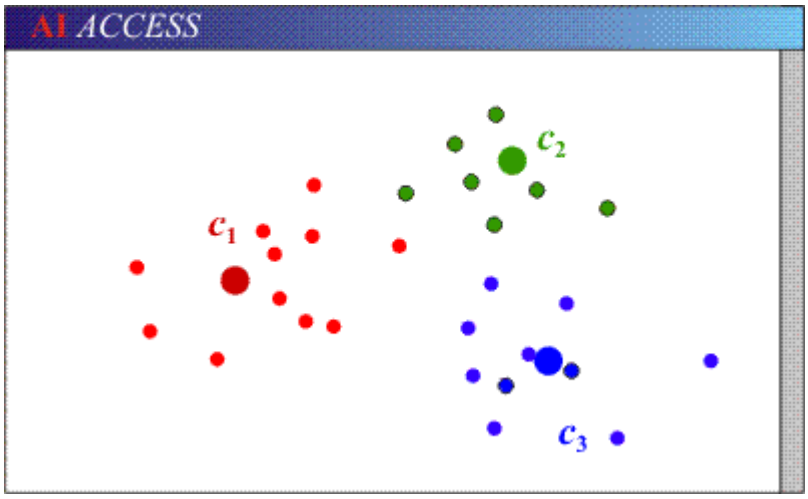
TP6 : Image segmentation using K-means

Sommaire

I) Objectif :

Le but de ce TP est de segmenter l'image afin de regrouper les pixels en plusieurs régions suivant des critères prédéfinis.
La méthode k-means permet ainsi de faire cela en partitionnant les pixels en k groupes selon des caractéristiques communes (positions, couleur). Ce résultat est obtenu en positionnant k « centroïdes » sur l'image. Chaque pixel est alors affectée au « centroïde » le plus proche (règle dite « de la Distance Minimale ») comme sur l'illustration ci-dessous :



Pour l'implémentation de la fonction K-means, nous avons écrit tout d'abord les fonctions suivantes :

- **distance3D** (pixel C, pixel B) : elle prend donc en paramètre 2 pixels et renvoie la distance en couleur entre ces 2 pixels. Un pixel possède deux champs : sa position (ligne, colonne) sur l'image et sa couleur (R,G,B).

- **distance5D** (pixel C, pixel B) : elle prend donc en paramètre 2 pixels et renvoie la distance en couleur et en pixels.

- **moyenne**(liste de pixels) : elle prend donc en paramètre une liste de pixels et renvoie un pixel qui a pour position la moyenne en position de la liste de pixels et pour couleur la moyenne en couleur de la liste de pixels.

- **InitCentroïdes** (int k) : cette fonction prend en paramètre un entier k représentant le nombre de « centroïdes ». Elle renvoie une liste de k pixels assez bien répartis sur l'image.

- **InitCentroïdesAleatoires** (int k, int seed) : Cette fonction prend en paramètre un entier k représentant le nombre de « centroïdes » et un entier seed qui est le germe de la fonction aléatoire. Elle renvoie une liste de k pixels récupérés aléatoirement sur l'image.

II) Applications

A) Images PGM (P2 ou P5)

Les exemples qui suivent ont été traités avec la distance n'évaluant que le niveau de gris.

- Exemple 1 : « test.pgm »

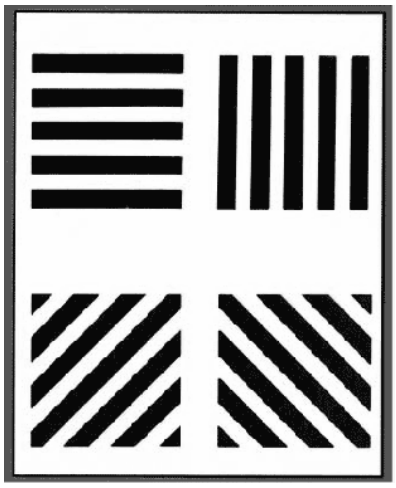


image originale

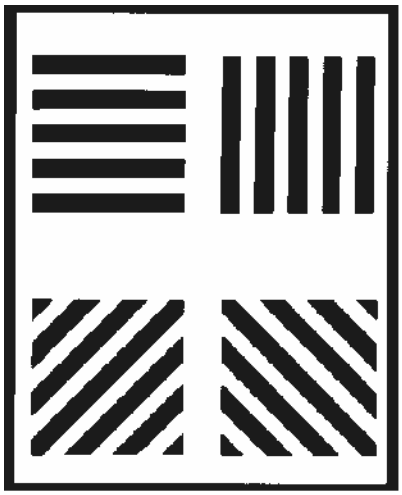


image segmentée avec k=2

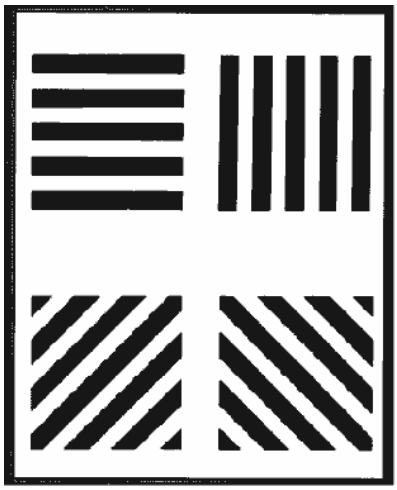


image segmentée avec k=4

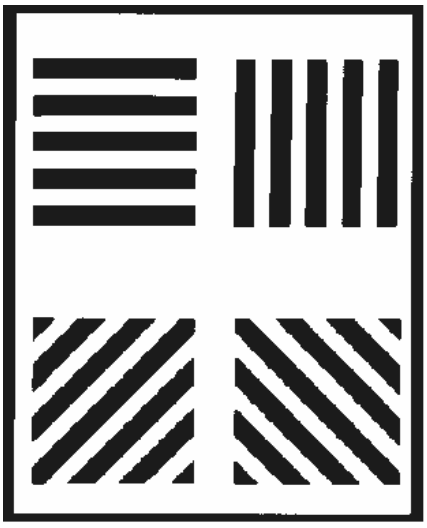


image segmentée avec k=5

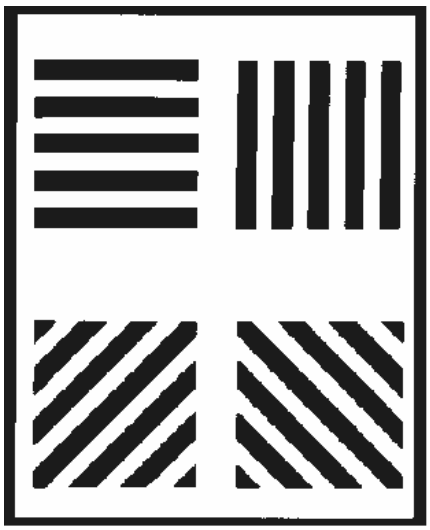


image segmentée avec k=6

Observations :

Cette image nous permet de tester notre programme, car elle ne comporte que deux couleurs qui sont bien segmentées par notre programme. Quelque soit le nombre de groupe (k >= 2), l'image est segmentée en deux parties.

- Exemple 2 : « fleurs.pgm »

image originale

image segmentée avec k=3

image segmentée avec k=4

image segmentée avec k=5

image segmentée avec k=6

Observations :

On peut voir que plus le k est grand et plus l'image segmentée se rapproche de l'image originale.

Ceci est cohérent, car plus il y a de groupes (non confondus), plus la palette des couleurs est grande et plus la représentation de la segmentation est riche en détails.

- Exemple 3 : « graines.pgm »

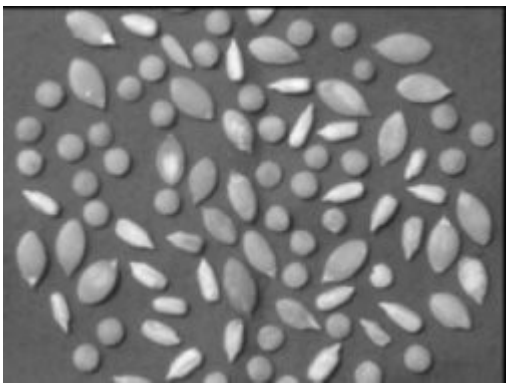


image originale

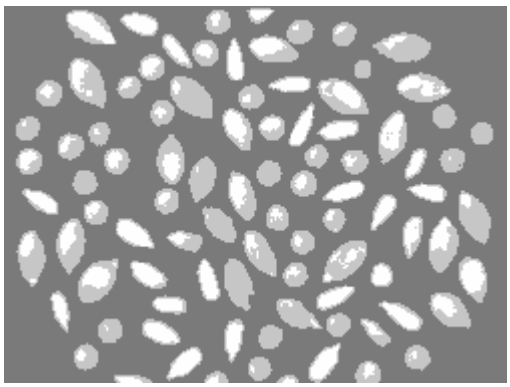


image segmentée avec k=3

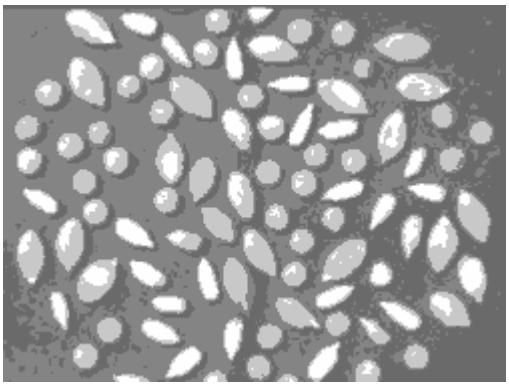


image segmentée avec k=4

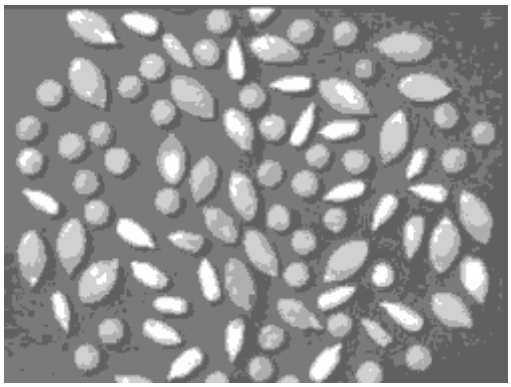


image segmentée avec k=5

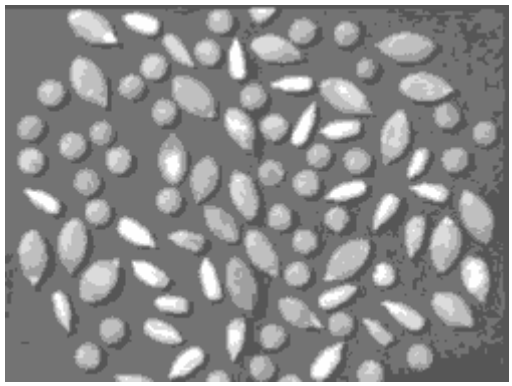


image segmentée avec k=6

Observations :

Même observation que l'exemple 1. Néanmoins, on remarque qu'à partir de k=4, l'image est assez proche de l'image originale.

B) Images PPM (P3 ou P6)

1) Distance3D (couleur)

Les exemples qui suivent ont été traités avec la distance n'évaluant que les couleurs (R,G,B).
Voici quelques tests qui illustrent la segmentation d'une image PPM dont le nombre de groupes k varie :

- Exemple 1 : « grenouille.ppm »

image originale

image segmentée avec k = 2

image segmentée avec k = 4

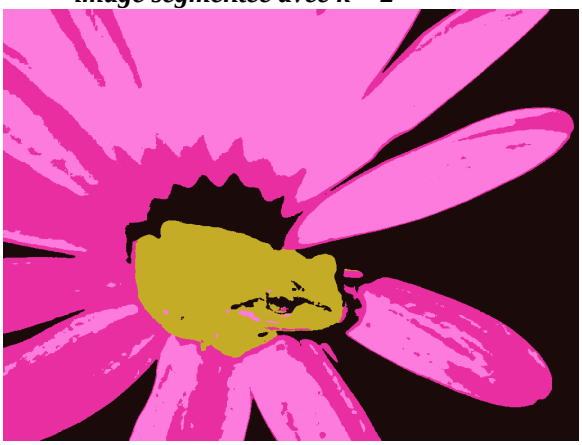


image segmentée avec k = 3

image segmentée avec k = 5

image segmentée avec k = 10

- Exemple 2 : « zebre.ppm »

image originale

image segmentée avec k = 2

image segmentée avec k = 3

image segmentée avec k = 4

image segmentée avec k = 5

image segmentée avec k = 10

Observations :

On observe encore que plus le nombre k de centroïdes augmente et plus l'image se rapproche de l'image originale.

Voici quelques tests qui illustrent la segmentation d'une image PPM dont les centroïdes sont placés à différentes positions (placement aléatoire). On affiche les centroïdes initiaux à l'aide d'une croix **rouge** :

- Exemple 1 : « grenouille.ppm »

image originale

image originale segmentée avec k = 4

image originale segmentée avec k = 4

image originale segmentée avec k = 4

image originale segmentée avec k = 4

- Exemple 2 : « zebre.ppm »

image originale

image originale segmentée avec k = 4

image originale segmentée avec k = 4

image originale segmentée avec k = 4

image originale segmentée avec k = 4

Observations :
On remarque que le placement des centroïdes a une grande influence sur la segmentation de l'image. Lorsqu'on place les centroïdes dans une même zone, celle-ci se retrouve plus partitionnée au détriment des autres zones qui elles sont uniformisées.

Voici quelques tests qui illustrent les différentes itérations de la segmentation d'une image PPM (avec k = 5 le nombre de centroïdes) :

- Exemple 1 : « grenouille.ppm »

image originale

image segmentée après 1 itération

image segmentée après 2 itérations

image segmentée après 3 itérations

image segmentée après 4 itérations

image segmentée après 5 itérations

image segmentée après 6 itérations

image segmentée après 7 itérations

image segmentée après 8 itérations

- Exemple 2 : « zebre.ppm »

image originale

image segmentée après 1 itération

image segmentée après 2 itérations

image segmentée après 3 itérations

image segmentée après 4 itérations

image segmentée après 5 itérations

image segmentée après 6 itérations

image segmentée après 7 itérations

image segmentée après 8 itérations

Observations :
A partir de la 8 ème itération, les régions semblent se fixer. On peut voir que plus il y a d'itérations, moins les régions sont dispersées.

2) Distance5D (couleur et position)

Voici quelques tests qui illustrent la segmentation d'une image PPM (avec k = 5 le nombre de centroïdes) suivant une distance en 5 dimensions (position (i,j) et couleur (R,G,B)) :

- Exemple 1 : « grenouille.ppm »

image originale

image segmentée après 1 itération

image segmentée après 3 itérations

image segmentée après 5 itérations

image segmentée après 8 itérations

- Exemple 2 : « zebres.ppm »

image originale

image segmentée après 1 itération

image segmentée après 3 itérations

image segmentée après 5 itérations

image segmentée après 8 itérations

Observations :

On remarque que la distance pixel influe grandement sur le partitionnement. L'image se retrouve découpée en morceaux qui dépendent de la position des pixels. C'est pour cela que les groupes sont plus compacts, ils sont moins dispersées.