

# INF555 TD5

## Vector Quantization: L'algorithme $k$ -Means

Frank NIELSEN  
nielsen@lix.polytechnique.fr

12 Octobre 2011

Nous nous proposons de convertir une image couleur (3 canaux de couleur RGB 8 bits chacun) en une image ayant seulement  $k$  couleurs distinctes. Ces  $k$  couleurs forment la palette des couleurs.

- Lire l'image couleur `polytechnique.png` et convertir chaque pixel en un point 3D équivalent  $(r, g, b)$  dans l'espace des couleurs.
- Implanter l'algorithme  $k$ -means pour trouver une table des couleurs (*colormap*) de taille  $k$  donné (essayez avec  $k = 256$  puis  $k = 8$ ):
  - On utilisera l'initialisation de Forgy qui consiste à tirer les graines (centroïdes initiaux) au hasard parmi le nuage de points.
  - On affichera à chaque itération la fonction de perte: C'est-à-dire la somme des distances *au carré* des points à leur plus proche centroïde. On vérifie que cette fonction de perte décroît de façon monotone.
- Grâce au codebook trouvé, générer l'image correspondante ayant seulement  $k$  couleurs: Chaque pixel sera converti dans sa couleur la plus proche. Observer le résultat de cette segmentation (les régions ne sont pas connexes dans l'image).
- Visualiser avec Processing/OpenGL le nuage de points 3D des couleurs des pixels et les différentes itérations de l'algorithme  $k$ -means.
- Utiliser l'initialisation de  $k$ -means++:  
<http://en.wikipedia.org/wiki/K-means%2B%2B>
- \*\* Utiliser  $k$ -means pour initialiser un algorithme de soft clustering: Expectation-maximization:  
[http://en.wikipedia.org/wiki/Expectation-maximization\\_algorithm](http://en.wikipedia.org/wiki/Expectation-maximization_algorithm)

