7 Quantification

Le module `image` de `matplotlib` permet d'ouvrir un fichier d'image au format 'png', sous la forme d'un `numpy.array`. La couleur de chaque pixel est représentée par un triplet de $[0,1]^3$ (format 'RGB').

```
import matplotlib.pyplot as plt
import matplotlib.image as img
import numpy as np

# reading png image file
im = img.imread('moissonneuse.png')
# show image
plt.imshow(im)
```

1. Ecrire un script où k couleurs $z_1, \dots, z_k$ sont choisies au hasard et l'image est modifiée de la façon suivante : la couleur de chaque pixel est changée par la couleur la plus proche dans l'ensemble $\{z_1, \dots, z_k\}$. Pour définir la notion de proximité on pourra utiliser la norme ℓ^p de $[0,1]^3$ donnée par $\|(u_1,u_2,u_2)\|_p = \left(\sum_{i=1}^3 |u_i|^p\right)^{1/p}$.
2. Pour un nombre k de couleurs donné, on cherche le "meilleur choix" de $z_1, \dots, z_k$, autrement dit on veut minimiser

$$\sum_x \min_{1\le i\le k} \|z(x)-z_i\|_p^q$$

où la somme est prise sur tous les pixels et $z(x)$ représente la couleur du pixel x. ($q>0$ est une constante)
Pour ce faire on peut mettre en œuvre l'algorithme de gradient stochastique suivant :
On se donne un pas $\gamma > 0$ "petit" et des valeurs initiales de $z_1, \dots, z_k$.
On répète autant de fois que possible :

(a) Choisir un pixel x au hasard

(b) Déterminer $i \in \{1, \dots, k\}$ tel que $\|z_i - z(x)\|_p = \min_\ell \|z_\ell - z(x)\|_p$

(c) $z_i \leftarrow z_i - \gamma \nabla_{z_i} \|z_i - z(x)\|_p^q$

Ici $\nabla_{z_i}\|z_i - z(x)\|_p^q$ désigne le gradient de la fonction $y \mapsto \|y - z(x)\|_p^q$ au point z_i. En particulier, pour $p=2$ et $q=2$ on a $\nabla_{z_i}\|z_i - z(x)\|_2^2 = 2(z_i - z(x))$.