

# TP1 - Codage et compression multimédia

## Transformation d'une image couleur (16 millions de couleurs) en une image couleur (palette couleurs)

---

*Léa Serrano M1 IMAGINE*

Lien de mon git pour ce tp : <https://github.com/LeaSerrano/M1-IMAGINE-CompressionDonnees-TP1.git>

### Table des matières

1	<a href="#">Ex 1</a>	2
2	<a href="#">Ex 2</a>	3
3	<a href="#">Ex 3</a>	3
4	<a href="#">Ex 4</a>	4

## 1 Ex 1

(questions 1 à 4 du tp)

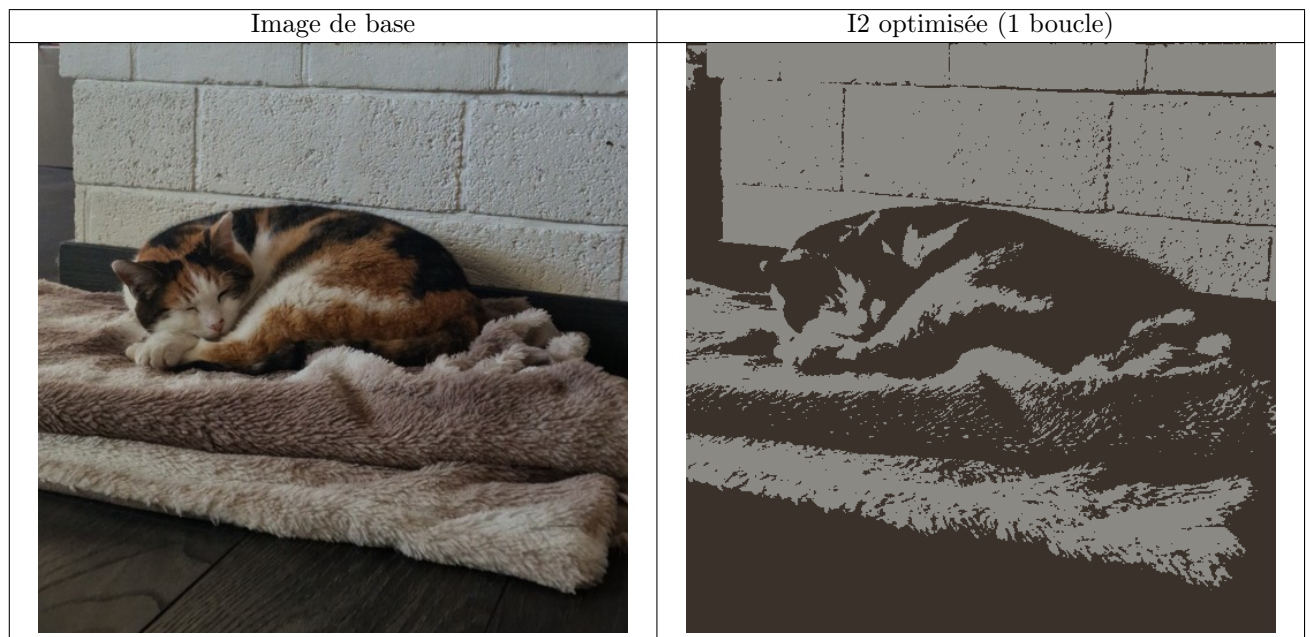
Dans ce premier exercice, le but va être de réaliser un algorithme de K-mean en utilisant  $K = 2$ .

On verra ensuite le résultat après une application de cet algorithme et le résultat après convergence (le plus optimisé).

Pour faire cela, on va d'abord sélectionner deux points sur l'image aléatoirement qui seront les centroïdes de notre image. Ensuite, on va parcourir les pixels de notre image et pour chaque pixel on va calculer la distance entre le pixel et les deux centroïdes et garder la distance minimale. Ensuite, on va calculer la moyenne de tous les pixels rattachés à chaque centroïde et leur faire prendre la valeur de la moyenne (le centroïde prendra aussi cette valeur).

Et on va répéter un certain nombre de fois jusqu'à ce que la différence entre un centroïde et son centroïde précédent soit petit (ici j'ai fixé la condition d'arrêt sur le fait que la différence soit inférieure à 10).

J'ai donc appliqué cet algorithme avec deux centroïdes aléatoires :


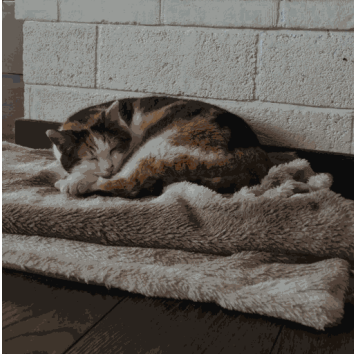
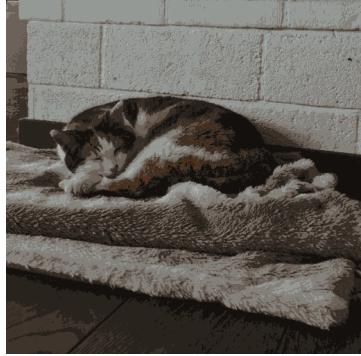





## 2 Ex 2

(questions 5 et 6 du tp)

On va maintenant réaliser la même chose mais cette fois-ci avec 256 couleurs.

Voici l'algorithme avec 256 couleurs et différents nombres de boucles :

Image de base	I256 1 boucle	I256 2 boucles
		
I256 3 boucles	I256 10 boucles	I256 optimisée (12 boucles)
		

On voit que les images sont assez différentes en fonction du nombre de boucles.

## 3 Ex 3

(question 7 du tp)

On va maintenant calculer le PSNR de notre image. Le PSNR se calcule de cette façon :

$$PSNR = 10 * \log_{10}(255^2 / EQM)$$

avec

$$EQM = \sum (p(i) - p'(i))^2$$

avec p(i) un pixel de l'image 1 et p'(i) un pixel de l'image 2.

On va prendre pour l'image 1, l'image de base.

Si on prend pour l'image 2, I256 1 boucle, le PSNR obtenu est de 31.0 dB.




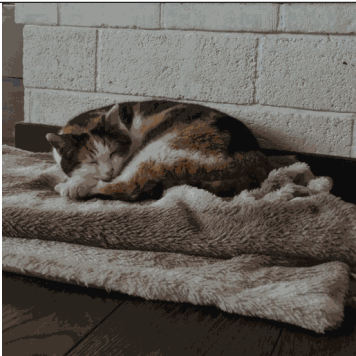
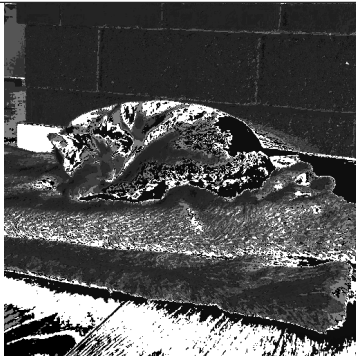
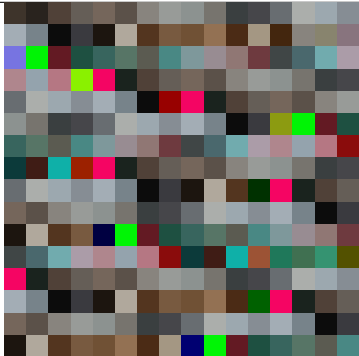

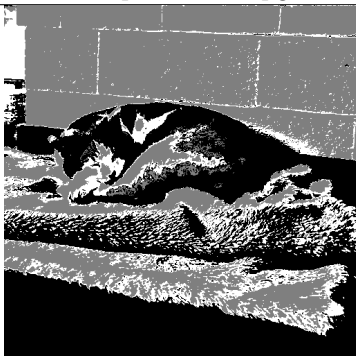
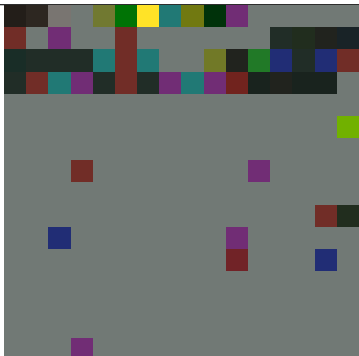
Si on prend pour l'image 2, I256 optimisée, le PSNR obtenu est de 17.7 dB.

## 4 Ex 4

(question 8 du tp)

On va ensuite réaliser une palette et transformer notre image en image en niveaux de gris.

Voici le résultat obtenu :

I2	I2 en pgm	Palette
		
I256 avec 1 boucle	I256 en pgm	Palette
		
I256 optimisée	I256 optimisée en pgm	Palette
		

On voit que la palette de l'image optimisée contient moins de nuances de couleurs que celle de l'image après une seule boucle. Cela est lié à l'algorithme de K-mean qui va regrouper des pixels en fonction de leur similarité de couleur.