

UNIVERSITÉ DE MONTPELLIER

---

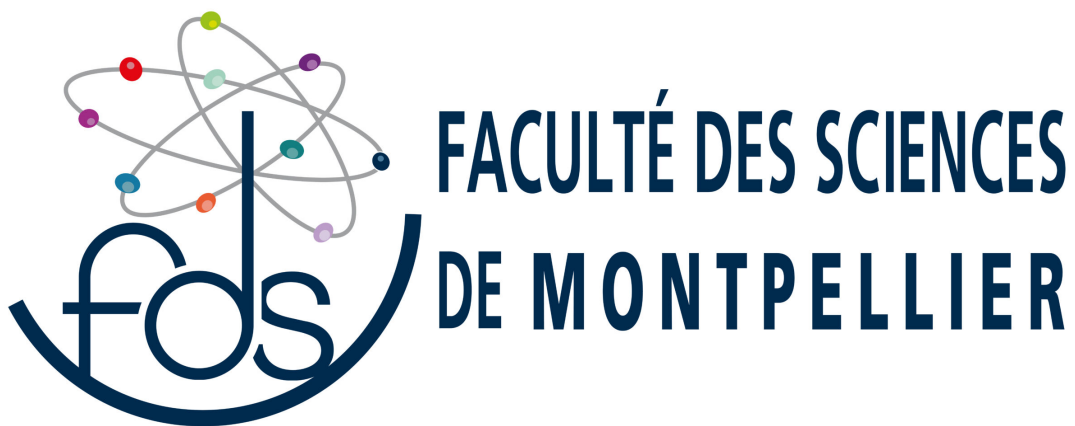
M1 IMAGINE - Codage et Compression  
Compte Rendu TP1 - Transformation d'image 16M  
Couleur vers palette de couleur

---

*Etudiant :*  
Guillaume Bataille

*Encadrant :*  
William PUECH

Année 2022-2023



## 0.1 Contexte et objectif

**Contexte** Dans le cours, nous avons revu les bases concernant les données multimédia, leurs différents support de stockage et aussi quelques manière de les représenter, notamment via des normes/format différents.

**Objectif :**

- Réduire le nombre de couleurs contenu dans une image couleur afin de réduire sa taille.
- Appliquer un algorithme de K-mean afin de réduire le nombre de couleurs.
- Générer une image d'index associée à une palette de couleurs.
- Observer l'impact sur la qualité d'une image couleur après réduction des couleurs.
- Programmer le tout pour avoir en entrée une image couleur au format ppm (avec 16 millions de couleurs) et en sortie une image au format pgm en y associant une palette de couleurs et calculer sa qualité en termes de PSNR

Voici l'image couleur (au format ppm) choisie pour nos différents traitement :



FIGURE 1 – Gecko.ppm (635x475)

C'est un lézard vert de manapany, une espèce endémique et protégée de l'île de la Réunion.

# Sommaire

0.1	Contexte et objectif . . . . .	1
<b>1</b>	<b>K-mean</b>	<b>3</b>
1.1	Fonctionnement . . . . .	3
1.2	K-mean basique de taille 2 . . . . .	3
<b>2</b>	<b>K-mean 4</b>	<b>4</b>
<b>3</b>	<b>K-mean 8</b>	<b>5</b>
<b>4</b>	<b>K-mean 16</b>	<b>6</b>
<b>5</b>	<b>K-mean 32</b>	<b>7</b>
<b>6</b>	<b>K-mean 64</b>	<b>8</b>
<b>7</b>	<b>K-mean 128</b>	<b>9</b>
<b>8</b>	<b>K-mean 256</b>	<b>10</b>
<b>9</b>	<b>Retour au niveau de gris pour une compression ppm -&gt; pgm + palette</b>	<b>11</b>
<b>10</b>	<b>Conclusion</b>	<b>12</b>

# 1. K-mean

## 1.1 Fonctionnement

Le K-mean est une façon de compresser une image dans le nombre d'intensité de couleur présente dans celle-ci. L'objectif étant de se ramener à un nombre de couleur restreint afin de résumer ces informations via des couleurs "classe", ce qui nous donne une palette de n couleurs.

Cela se base sur un algorithme récursif :

- Trouver n couleurs (classe) assez distante dans l'espace couleur RGB de mon image initiale
- Parcourir tout les pixels de mon image et les ranger avec la classe la plus proche.
- Mettre à jour la classe représentant par la moyenne de tout les pixels sous-jacent.

Et on boucle récursivement jusqu'à une certaine profondeur (donnée arbitrairement) ou lorsque la couleur de l'ancienne couleur classe et la nouvelle couleur classe n'est qu'à un petit epsilon de différence.

**Au final** , on se retrouve avec une convergence de n couleur représentant notre image via une palette de n couleur différente. Cela est un gain considérable (16 Million de couleur initialement) surtout que visuellement, il est difficile de voir la différence à l'oeil nu.

## 1.2 K-mean basique de taille 2

Voici une application du K-mean 2 : Le PSNR de cette nouvelle image K-mean est de **21,4** dB.

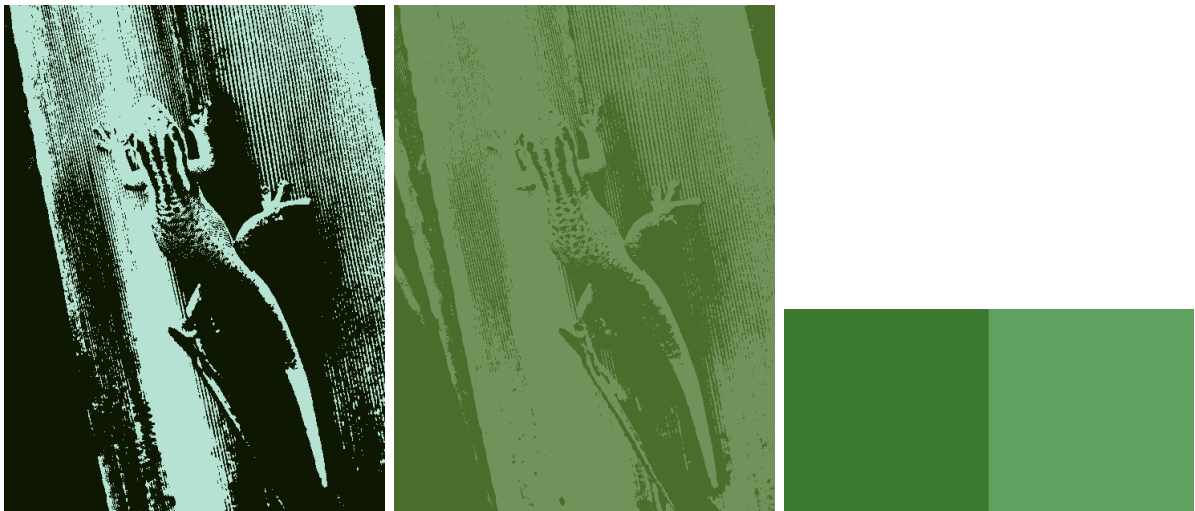


FIGURE 1.1 – Kmean 2 de gecko.ppm avant moyennage - après moyennage - palette utilisées

## 2. K-mean 4

Pour les autres versions, on augmente le nombre de classe et donc le nombre de couleur utilisées pour la palette.

Et plus on augmente cette valeur, plus on se rapproche visuellement de l'image initiale.

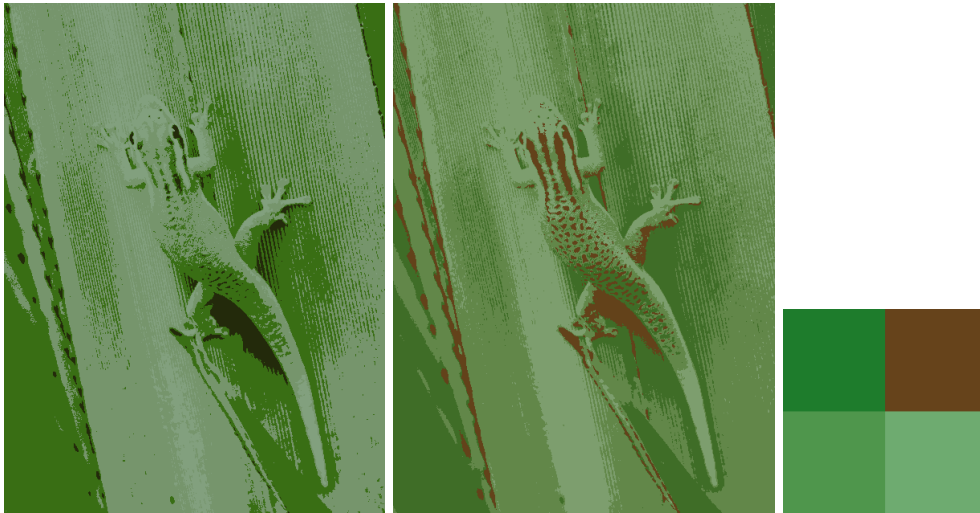


FIGURE 2.1 – Kmean 4 de gecko.ppm avant moyennage - après moyennage - palette utilisées

Le PSNR de cette nouvelle image K-mean est de **23,8** dB.

### 3. K-mean 8



FIGURE 3.1 – Kmean 8 de gecko.ppm avant moyennage - après moyennage - palette utilisées

Le PSNR de cette nouvelle image K-mean est de **25,7** dB.

## 4. K-mean 16



FIGURE 4.1 – Kmean 16 de gecko.ppm avant moyennage - après moyennage - palette utilisées

Le PSNR de cette nouvelle image K-mean est de **27,63** dB.

## 5. K-mean 32



FIGURE 5.1 – Kmean 32 de gecko.ppm avant moyennage - après moyennage - palette utilisées

Le PSNR de cette nouvelle image K-mean est de **30,2** dB.



## 6. K-mean 64



FIGURE 6.1 – Kmean 64 de gecko.ppm avant moyennage - après moyennage - palette utilisées

Le PSNR de cette nouvelle image K-mean est de **32,1** dB.

## 7. K-mean 128



FIGURE 7.1 – Kmean 128 de gecko.ppm avant moyennage - après moyennage - palette utilisés

Le PSNR de cette nouvelle image K-mean est de **34,0** dB.

## 8. K-mean 256



FIGURE 8.1 – Kmean 256 de gecko.ppm avant moyennage - après moyennage - palette utilisés

Le PSNR de cette nouvelle image K-mean est de **36** dB.

## 9. Retour au niveau de gris pour une compression ppm -> pgm + palette

Afin d'utiliser intelligemment cette palette de 256 couleurs, on décide de transformer notre image ppm au format pgm avec chaque couleur de la palette représenté par son id de la classe dans la palette (la première couleur de la palette d'id 0 va donner un niveau de gris 0, etc...). Cela nous donne donc façon de représenter une image ppm avec une image pgm bien plus légère et une palette de 256 couleur à ré-associer.

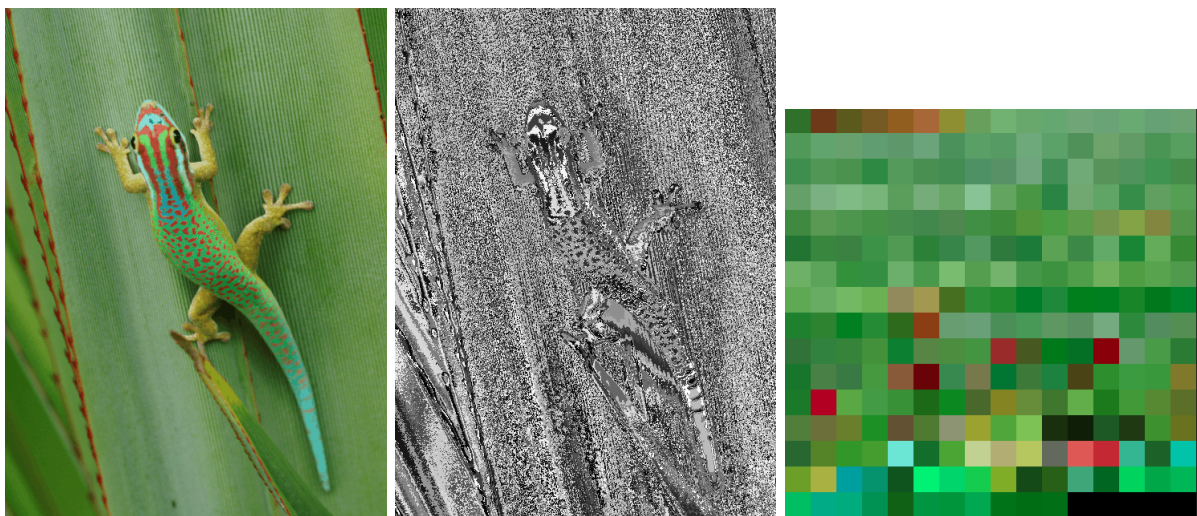


FIGURE 9.1 – Kmean 256 de gecko.ppm après moyennage - image de niveau de gris - palette utilisés

Avec respectivement les poids en KB :

Image initiale .ppm : **904,9 kB**

Image en niveau de gris .pgm + palette : **385,4 kB + 256\*3\*Octet**

Soit une compression de presque **2** avec une perte quasi négligeable de qualité pour l'oeil humain.

## 10. Conclusion

Nous avons donc réussi une compression de presque 2 en resumant les couleurs a une palette plus restreinte de couleur via recursion d'un K-mean. Ce qui peut être intéressant dans le partage et la simplification d'image.