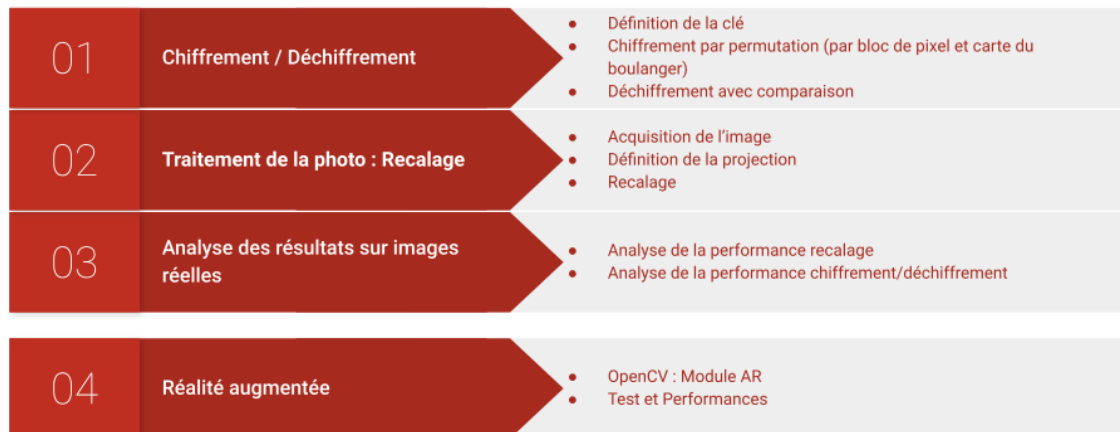


## CR 4 : Musée Sécurisé Virtuel

Semaine 15 au 19 novembre 2021

### Contexte

On cherche à chiffrer puis acquérir et déchiffrer des images, correspondant à des œuvres dans un musée.



### Chiffrement par permutation

On décide d'implémenter une première méthode de permutation que l'on appellera méthode naïve. Cette méthode consiste à découper l'image par bloc de X pixels puis à permuter ces blocs de blocs de pixels grâce à notre GPNA.

Inputs:

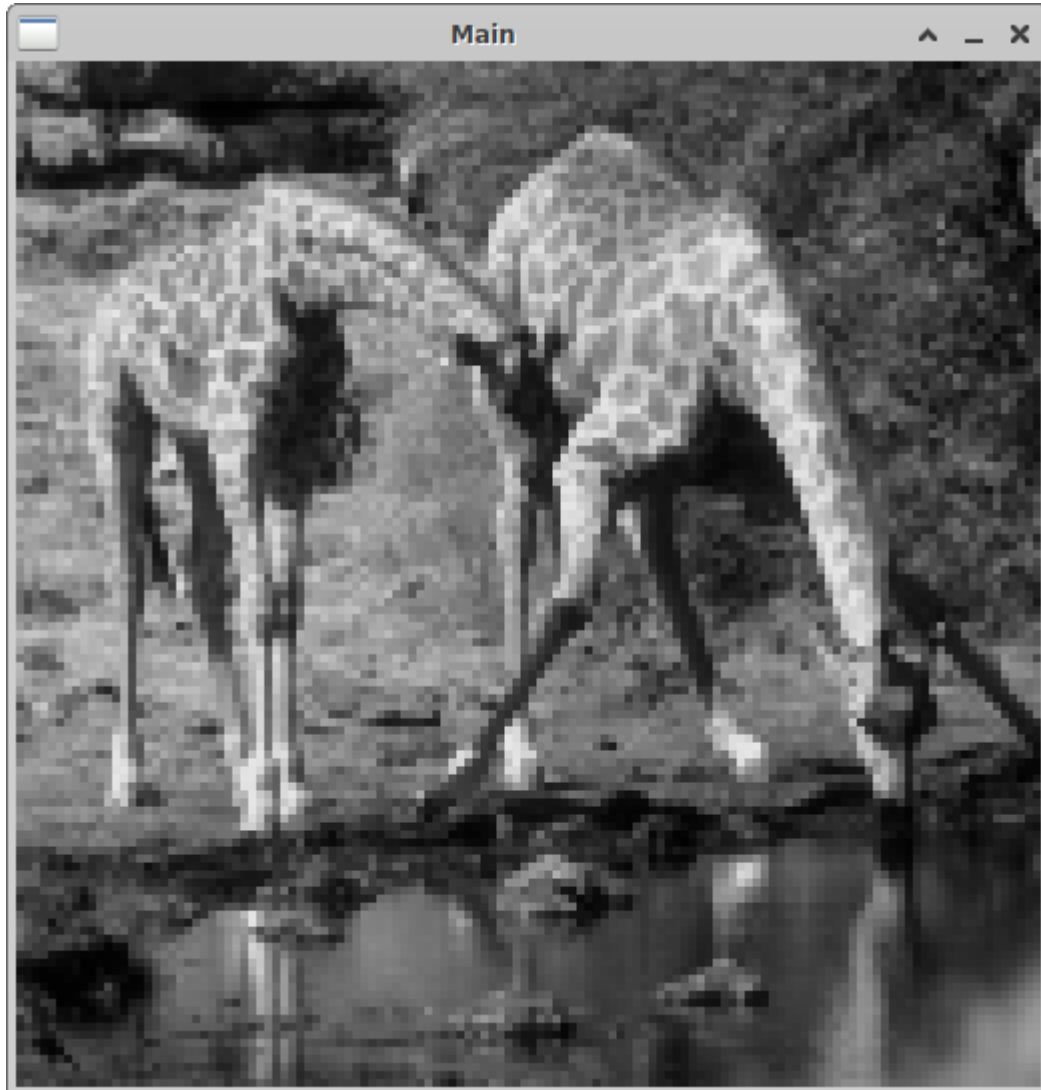
- Image I de taille 512x512 => on prendra par la suite une image carré de taille  $2^N$
- GPNA
- Taille du bloc de pixels = puissance de 2

Exemple de la chaîne pour un bloc de de 16 pixels (4x4)

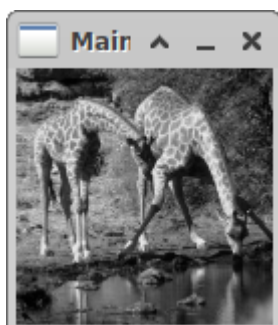
1/ Image de Base



2 / Image par block (mean)



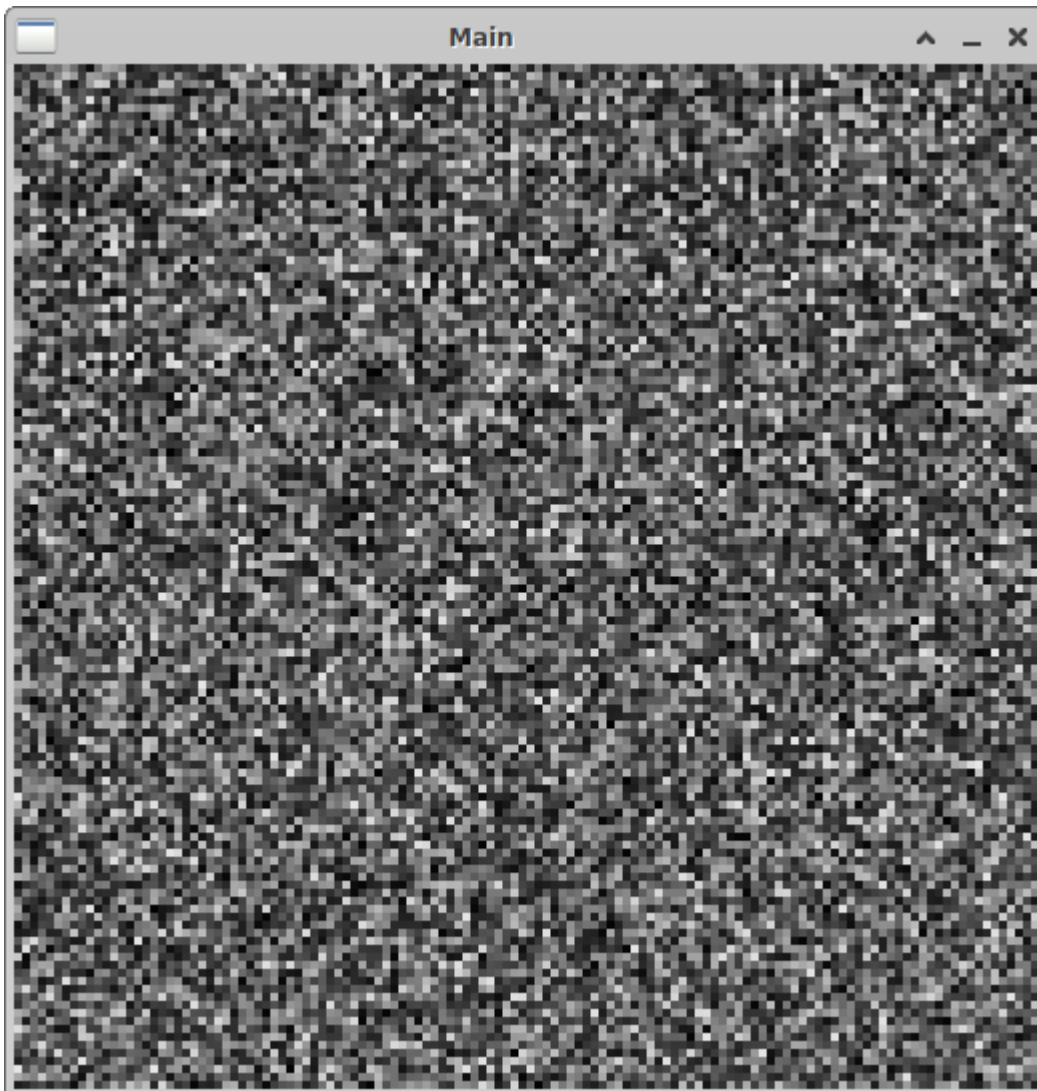
3 / Image subsampled (1 point par carré  $\sqrt{\text{taille de bloc}} \times \sqrt{\text{taille de bloc}}$ )



4/ Image subsampled permutée



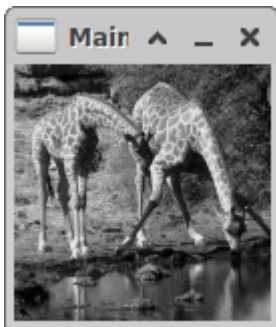
5/ upsampled image (Oeuvre chiffrée)



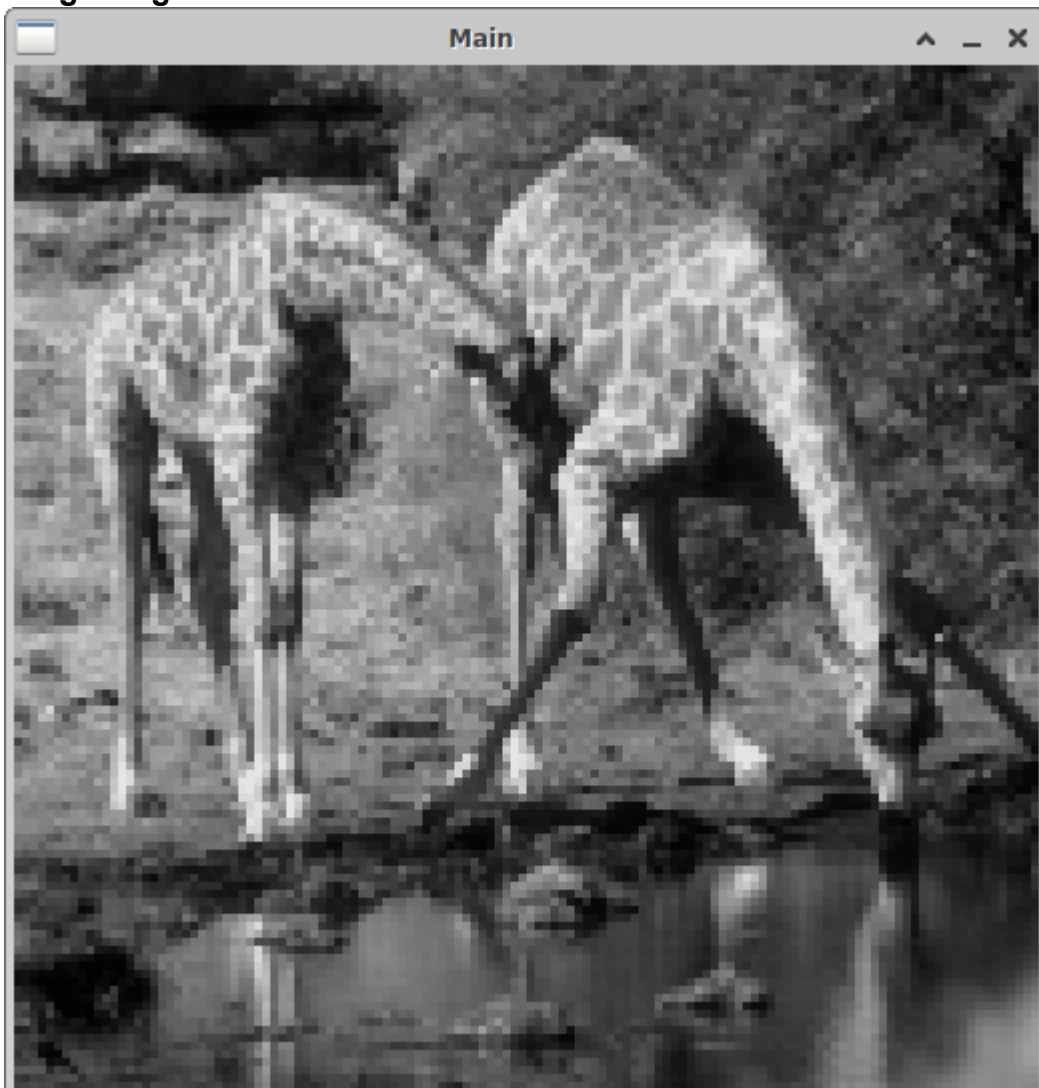
6/ oeuvre chiffrée subsampled



7/ subsampled image reconstruite



8/ Image Originale reconstruite

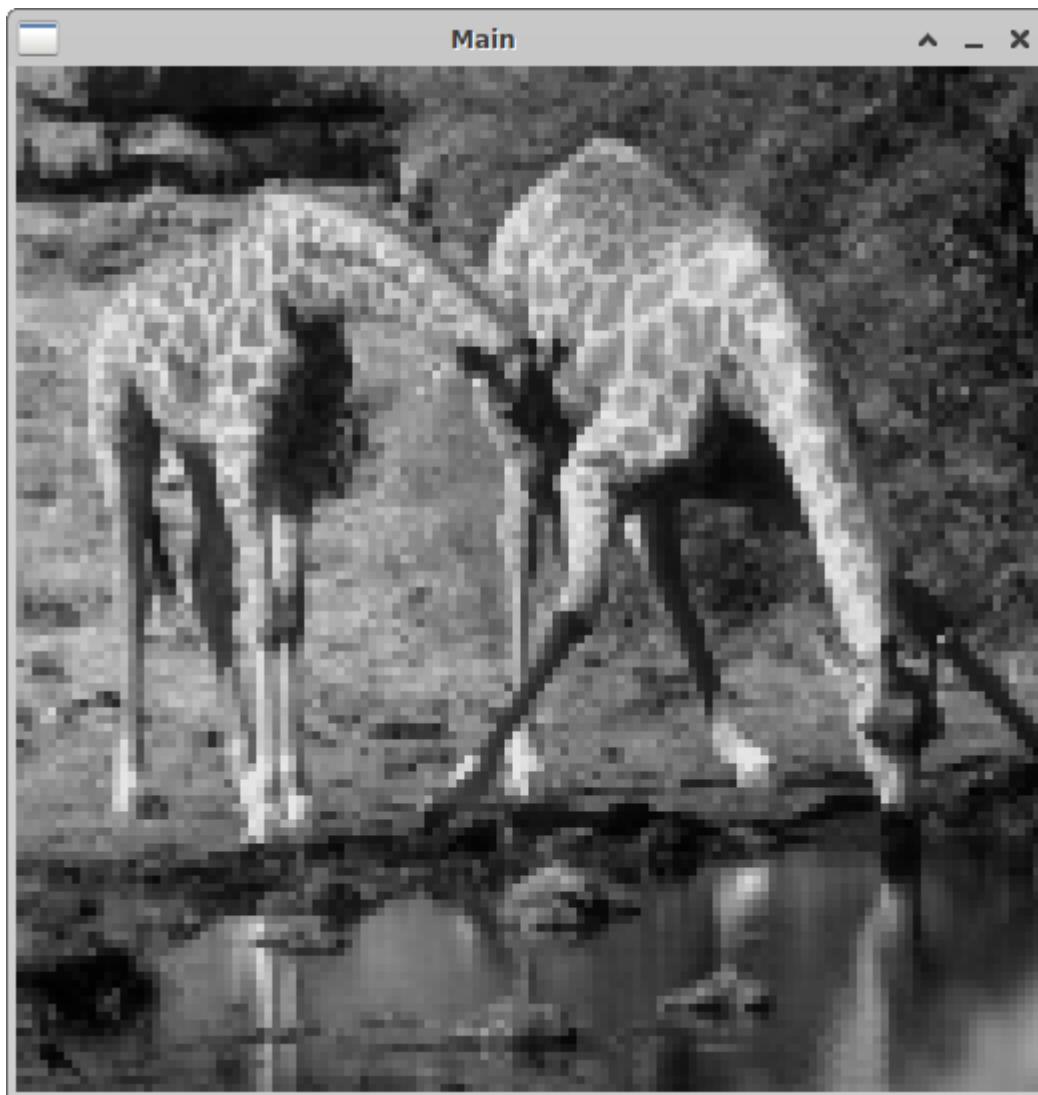


## PSNR

On calcule le PSNR sur notre image reconstruite (celle qui va apparaitre sur l'écran de l'utilisateur) versus (a) l'image d'origine et (b) l'image par block .

- (a) PSNR IMAGE original versus blocky = 21.7556,  
Min MSE=0 , Max MSE=36481
- (b) PSNR IMAGE on blocky = inf

*Image sur l'écran de l'utilisateur*

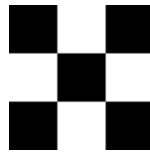


## Détection d'image

Notre œuvre telle que décrite dans le point 6, une fois chiffrée sera très texturée. La recherche de points d'intérêts directement sur notre image risque de ne pas donner des points d'intérêt spécifiques, voire pire des points de l'image.

Pour détecter l'image on cherche donc à détecter ses coins, pour cela on ajoute un pattern en damier aux 4 coins de l'image. Ce pattern permettra d'avoir des corners points plus robustes.

*Pattern*



**Détection du pattern :** Une fois les coins du pattern détectés, nous cherchons à retrouver le coin qui touche le coin de l'image. Pour cela ils nous faut déterminer le pattern auquel chaque point d'intérêts appartient, on fera un Hierarchical clustering.

Input : distance max intra class = X % de l'image

Arrêt: plus de fusion de classe

Ce critère nous permettra d'éliminer les outliers: classe avec moins de N points

CornerPoints via Harris [threshold] => [Hierarchical Clustering & Suppression des outliers]

Par la suite il nous restera 4 classes, chaque classe correspondant à un pattern et son centre de gravité est théoriquement le centre du pattern.

*Exemple d'image où la détection devra être faite*

