

HAI927I
Projet Image

Compte Rendu n°4
Sujet n°3 - Musée Sécurisé Virtuel

COUNILLE Alexandra
&
LIN-WEE-KUAN Malika



FACULTÉ DES SCIENCES
MONTPELLIER



UNIVERSITÉ
DE MONTPELLIER

Algorithme de chiffrement/déchiffrement par permutation

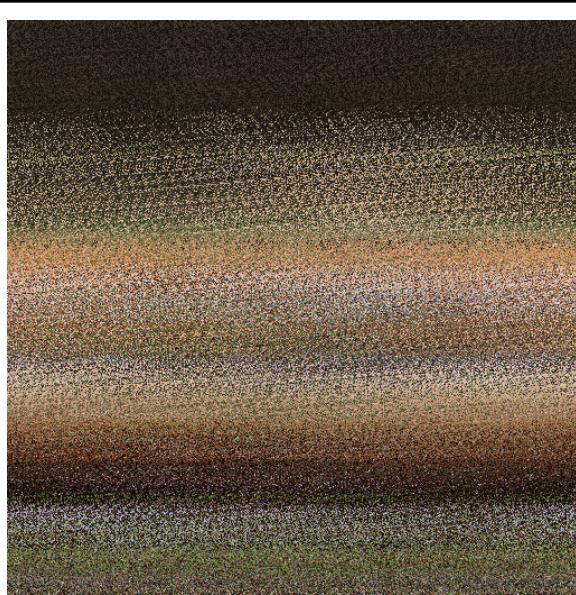
Méthode d'échanges des indices des symboles



Chiffrement avec une clé de taille 500.



Déchiffrement de l'image précédente.



Chiffrement avec une clé de taille 500.



Déchiffrement de l'image précédente.

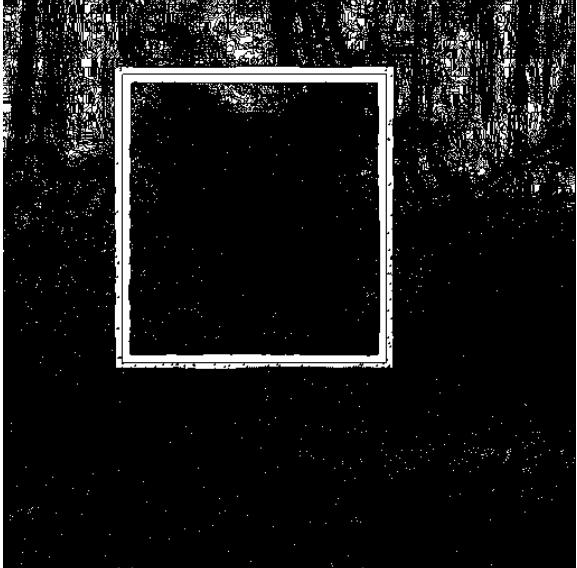
Nous avons amélioré notre méthode de chiffrement en permutant les pixels à la chaîne plutôt que de permutez les lignes, cela permet un meilleur chiffrement plus difficile à discerner. Et d'éviter les différents bugs de la méthode précédente.

La clé de permutation est générée pseudo-aléatoirement à l'aide d'une seed que l'on conservera pour retrouver ladite clé.

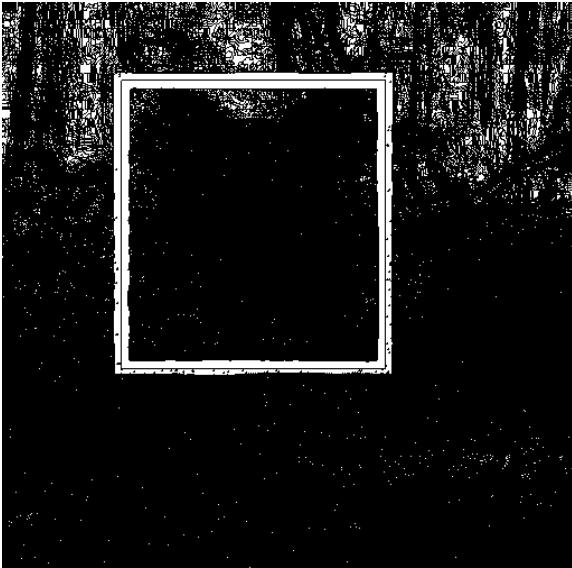
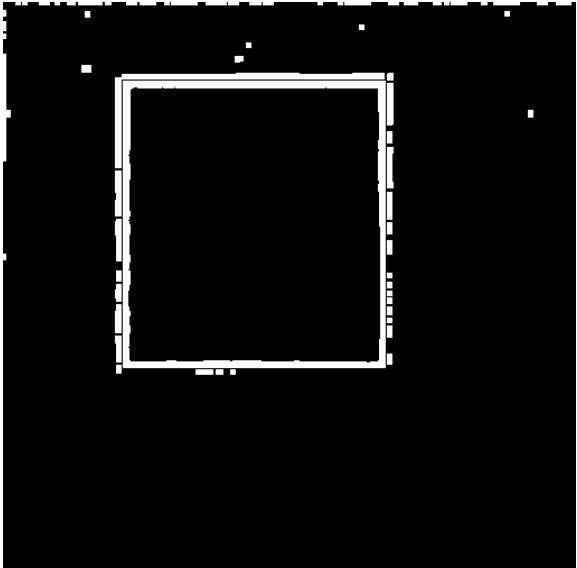
Algorithme de détection des contours

Calcul du gradient

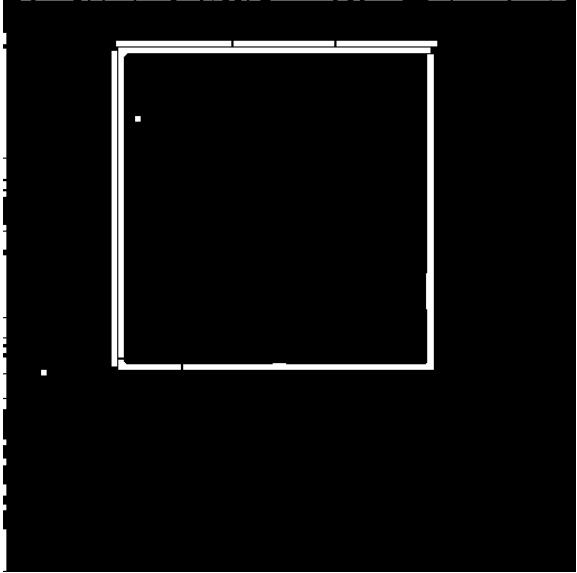
Avec le cadre noir entouré d'un contours blanc (qui simulerait le bord d'une feuille de papier blanche), la détection de contours se fait plus efficacement.

| | |
|--|---|
|  |  |
| Image contenant un cadre | Résultat de sa dérivée en binaire |

Le processus de fermeture est de ce fait plus efficace également. Le bruit est considérablement réduit.

| | |
|---|--|
|  |  |
| Dérivée de l'image | Fermeture appliquée à la dérivée |

Autre résultat :

| | |
|---|--|
|  |  |
| Image contenant un cadre | Fermeture appliquée à sa dérivée |

La prochaine étape est de récupérer le contenu à l'intérieur de ce cadre.

Transformations géométriques

L'image sur la photo ne sera pas forcément bien droite face à l'objectif, c'est pourquoi il sera parfois nécessaire d'appliquer des transformations géométriques afin de la recaler.

Rotation : chaque position des pixels $p(i,j)$ de l'image est multiplié par la matrice de rotation suivante :

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix}$$

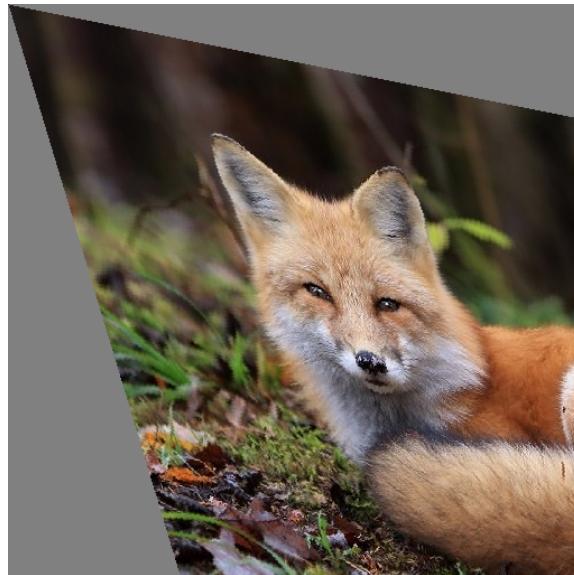


Rotation d'une image tournée de 60°

Déformation linéaire: Cette opération permet un recalage au niveau de la perspective. Le calcul de la position des pixels est le suivant :

$$x' = x + \alpha.y$$

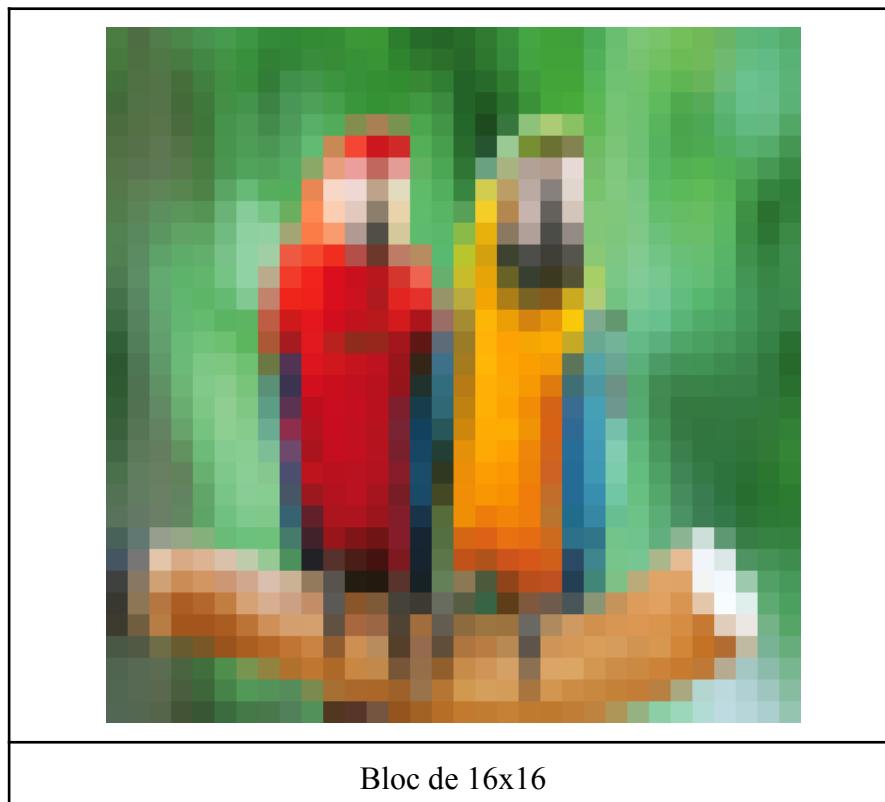
$$y' = y + \beta.x$$

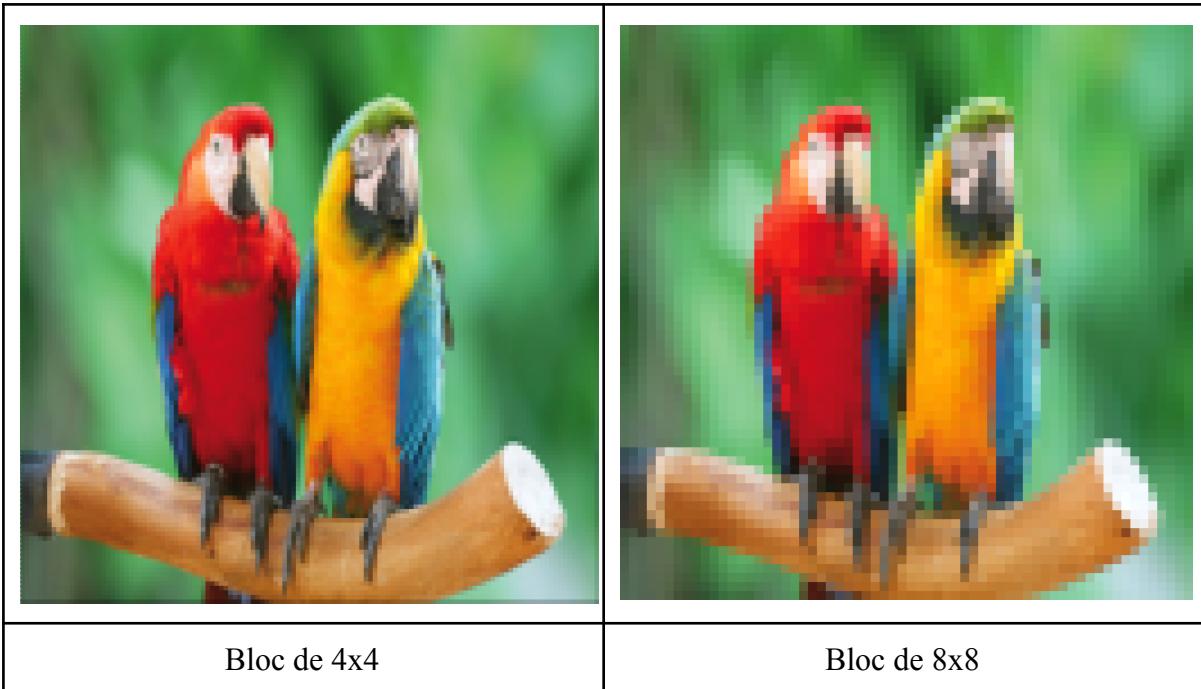


Exemple d'une déformation

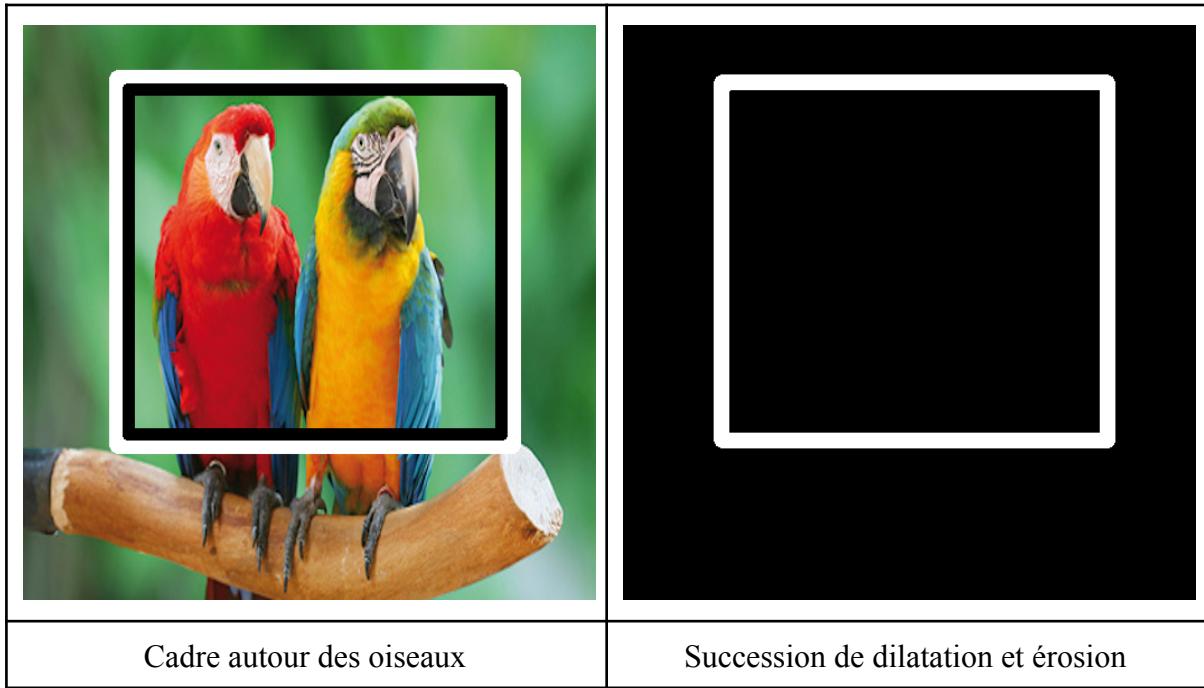
Conversion des pixels en bloc de 8x8

En faisant la moyenne des pixels voisins, sans changer la taille de l'image, on a fait des blocs de 4x4 et de 8x8 et de 16x16.





Récupération de l'image à l'intérieur du cadre



Avec un cadre plus épais et en appliquant manuellement des dilatations et érosions j'ai pu récupérer un cadre en binaire plus propre pour commencer à récupérer l'image à l'intérieur. En comparant les deux images ci-dessus, on a réussi à enlever les bords du haut et du bas (voir image ci-dessous) il manque donc les côtés pour terminer la récupération de l'image à l'intérieur d'un cadre.



Essais de récupération de l'image

Travail prévu pour la semaine du 22/11 au 28/11

- Terminer la récupération de l'image dans le cadre ;
- Assemblage de tous les algorithmes et application de ceux-ci ;
- Détection et calcul des coefficients de transformation pour le recalage.

Sources

- Transformations géométriques :
 - http://obligement.free.fr/articles/traitement_images_4.php
 - <https://images.math.cnrs.fr/Rotations-discretes.html>
 - <https://images.icube.unistra.fr/index.php/Fichier:4-Recalage-utf8.pdf>
 - <http://webia.lip6.fr/~thomen/Teaching/BIMA/cours/amelioration.pdf>