



**Année universitaire
2021 - 2022**

Majeure IMI — Partie 3 — 5ETI
**Compression et techniques avancées en
image**

TP de Tatouage d'images



**Marion Foare
Eric Van Reeth**

Présentation du TP

Objectifs

Après avoir vu en cours le processus d'insertion d'un tatouage dans le domaine de la DCT, il s'agira lors de ce TP d'insérer un **tatouage invisible et robuste dans l'espace de Fourier selon le même principe**. Par la suite, il s'agira d'implémenter une fonction de décodage capable de détecter la présence (ou non) du tatouage attendu, et d'évaluer la robustesse de la méthode vis-à-vis de différentes attaques.

Mise en place

Ce TP s'effectue en binôme par poste informatique (sous LINUX).

Le langage utilisé sera Python, avec la librairie suivante :

```
import numpy
from matplotlib import pyplot as plt
```

L'utilisation d'une IDE est fortement conseillée pour faciliter l'implémentation et le debug des codes. Il est également possible d'implémenter votre TP sous le format Jupyter Notebook, en sauvegardant votre script au format `.ipynb`. Veuillez bien à utiliser `Python3`, et non pas `Python2.7`. Pour cela, veuillez à ce que votre IDE utilise l'environnement Python adéquat : `/opt/python/cpe/bin/python3`

Évaluation

Dans un délai d'une semaine après la séance, vous rendrez (par binôme) :

- un **compte-rendu au format pdf ou jupyter notebook**, contenant les réponses aux questions, résultats et interprétations
- **vos code Python** fonctionnel et commenté de sorte à ce qu'il puisse être exécuté par le correcteur

1 Insertion du tatouage

Question 1 : Étant donné que l'image utilisée est à coefficients réels, rappelez les propriétés de symétrie de sa transformée de Fourier.

Question 2 : Lire l'image *cameraman.tif* et calculer sa transformée de Fourier discrète à deux dimensions.

Vérifiez la propriété citée plus haut.

Affichez son module en échelle logarithmique.

Question 3 : Générez un tatouage T , sous forme de vecteur contenant $K = 1024$ valeurs aléatoires, de moyenne nulle et d'écart-type 1.

Question 4 : Comme dans le cas de l'insertion de tatouages dans le domaine de la DCT, le tatouage est inséré sur les coefficients de Fourier d'amplitude élevée.

1. Rappelez pourquoi il est intéressant d'insérer le tatouage sur les coefficients de forte amplitude
2. Rappelez où sont situés en général les coefficients de forte amplitude dans l'espace de Fourier. S'agit-il des hautes ou des basses fréquences ?

3. En respectant toujours la propriété de la transformée de Fourier citée plus haut, proposez une stratégie d'insertion du tatouage sur les coefficients de Fourier. Implémenter l'insertion de T dans une fonction à part, qui prendra au minimum en entrée :

- la TFD-2D à tatouer
- le tatouage T à insérer
- le coefficient α de pondération du tatouage

Question 5 : On cherche maintenant à évaluer à quel point l'insertion du tatouage a dégradé l'image originale.

1. Pour la même instance du tatouage T , tracez l'évolution du PSNR résultant (en dB) en fonction de α :

$$PSNR(\alpha) = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE(\alpha)} \right) \quad (1)$$

Ici, MAX_I correspond au maximum d'intensité que peuvent prendre les pixels de l'image (255 sur 8 bits), et MSE l'erreur quadratique moyenne entre l'image originale et l'image tatouée : $MSE(\alpha) = \frac{1}{N} \sum_1^N (I[i] - I_{T_\alpha}[i])^2$.

2. Étant donné que l'on cherche à insérer un tatouage invisible, déterminer en justifiant votre réponse, une valeur de α qu'il vous paraît pertinent d'utiliser.

2 Détection du tatouage

Question 6 : En s'inspirant de l'exemple du tatouage sur la DCT, implémentez l'étape de détection du tatouage dans une fonction à part.

Cette fonction prendra en entrée (entre autres) un seuil permettant de décider si le tatouage détecté correspond à T . Décrivez la démarche suivie et les points clés de l'implémentation.

Question 7 : Vérifier le bon fonctionnement de la détection du tatouage T , en insérant en entrée de votre fonction une image contenant un autre tatouage T_b (et/ou l'image sans tatouage).

Question 8 : Vérifier la robustesse de votre méthode d'insertion vis-à-vis de plusieurs types d'attaques de votre choix (compression/décompression, rotation, translation, opération sur l'histogramme, ...). Analyser et discuter ces résultats. Conclure sur la méthode implémentée.

Annexe : Commandes Python

Commandes utiles

- Lecture d'une image : `I = plt.imread('cameraman.tif')`
- TFD-2D : `np.fft.fft2()`
- Recentrage de la TFD-2D : `np.fft.fftshift()`
- Génération d'un vecteur de valeurs aléatoires : `np.random.randn()`

Affichage d'une image

L'affichage d'images sera fait avec la librairie `matplotlib.pyplot` importée en tant que `plt` :

```
plt.figure()
plt.imshow(img, cmap='gray') #colormap gray
plt.title('My title')
plt.show()
```

L'affichage d'une image dans une partie **markdown** d'un Jupyter Notebook se fait via la commande suivante :

```
![Texte de remplacement](ImagePath.png "Titre de l'image")
```

Écriture d'une image

La fonction suivante permet d'enregistrer l'image courante (avec des options sur la qualité de compression jpeg notamment): `plt.savefig()`

Affichage en « subplot »

L'affichage de 3 images dans la même figure est effectué grâce à la commande `subplot` :

```
plt.subplot(131)
plt.imshow(img1, 'binary') #colormap 'binary'
plt.title('Binary Image')
plt.subplot(132)
plt.imshow(img2, 'gray') #colormap 'gray'
plt.title('Gray level')
plt.subplot(133)
plt.imshow(img3, 'jet') #colormap 'jet'
plt.title('Colored image')
plt.show()
```