# TPs de Vision

***Elio Genson***

Table des matières

[TPs de Vision 1](#_Toc8462)

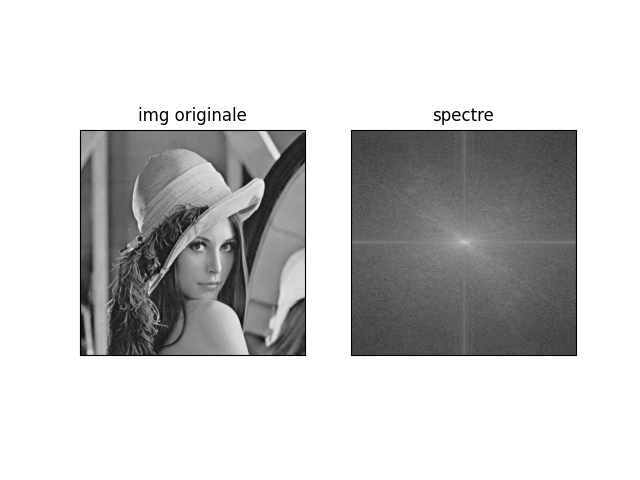
[TP1 2](#_Toc2725)

[1.Filtrage fréquentiel 2](#_Toc14634)

## TP1

### I.Filtrage fréquentiel

1. Image du spectre de Fourier :

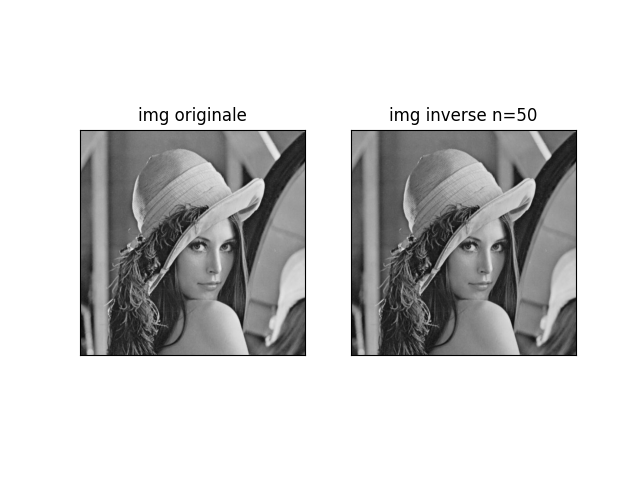


1. Image reconstruite :



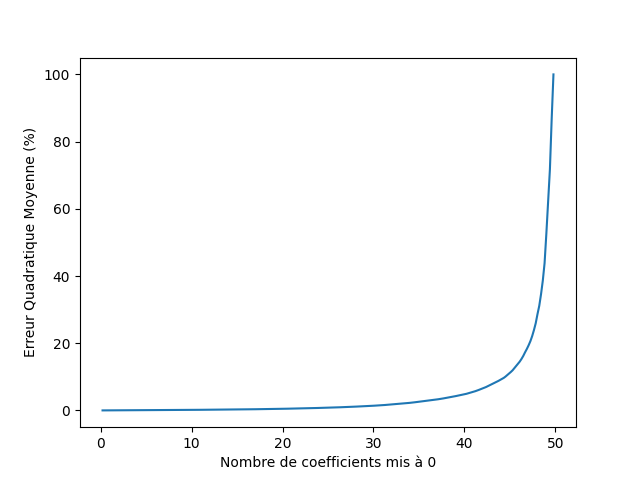
Le MSE calculé entre les deux images est de 17553

1. Images avec coefficient à 0 :



On remarque que l’image n=200 est bien détérioré, il y a des «vagues» sur les contours des formes.

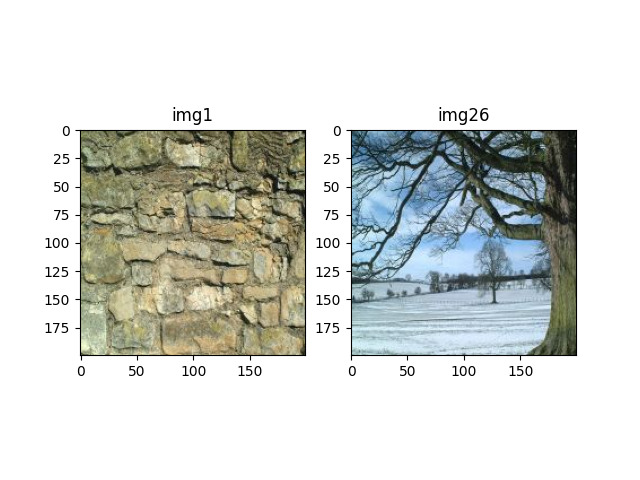
1. Graphe de la MSE en fonction du nombre de coef mit à zero :



### II.Caractérisation

Pour l’image 1 :

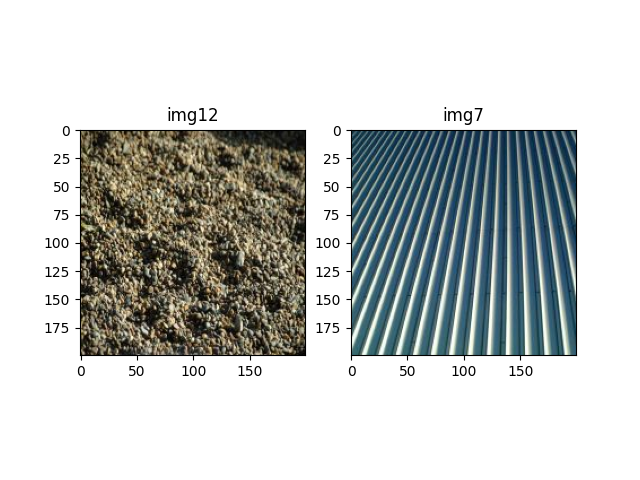
Distance de manhantan min = 249



Autres testes :

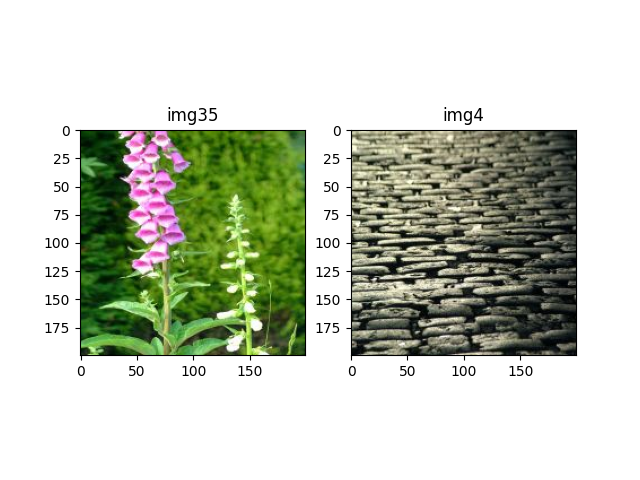
Img 12

Distance min = 35913



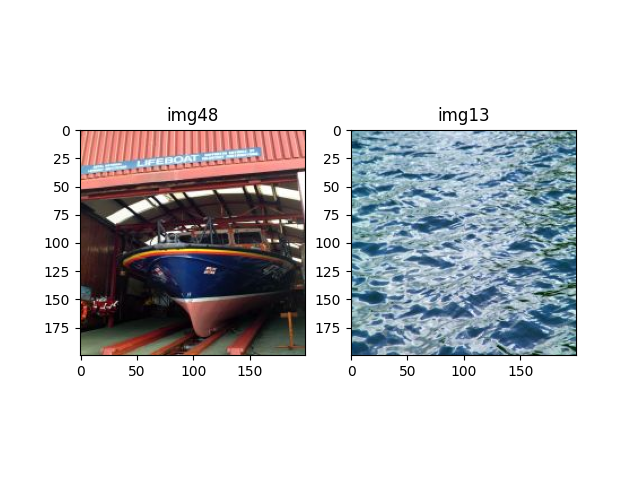
Img 35

Distance min = 9658



Img 48

Distance min = 8385



Je ne retrouve pas beaucoup de corrélation au niveaux des textures des images testés.

les seuls images qui on un rapport son la 35 et la 4 ou les feuilles on le même style de texture que les pavets.

## TP2

### SIFTs



Description du code :

cv.SIFT\_create() crée un détecteur SIFT (Scale-Invariant Feature Transform).

detectAndCompute(img, None) détecte les points d’intérêt (keypoints) et calcule leurs descripteurs.

k\_1, k\_2 : listes des points clés détectés.

des\_1, des\_2 : matrices contenant les descripteurs SIFT des points clés.

cv.BFMatcher (Brute-Force Matcher) est un algorithme d'appariement descripteur par descripteur.

bf.match(des\_1, des\_2) apparie les descripteurs des deux images.

sorted(matches, key=lambda x: x.distance) trie les correspondances par distance croissante (les meilleures correspondances en premier).