Transformée discrète en ondelettes : travaux pratiques

Yassine Jamoud, Samy Haffoudhi

24 octobre 2021

Introduction

Le but de ce TP est de prendre en main la transformée en ondelettes à l'aide de Matlab et de la boîte à outils `Wavelab`. Nous allons alors commencer par tracer des fonctions ondelettes, échelles et une transformée en ondelettes. Nous implémenterons ensuite une procédure de "débruitage" basée sur la transformée en ondelettes et enfin nous réaliserons de la compression d'images.

1 Tracé d'ondelettes et de fonctions échelles par DWT inverse

1. Sous forme informatique la DWT est représentée sous la forme :

   $$\texttt{DWT}(x) = [a_J, d_J, d_{J-1}, \ldots, d_1]$$

   où J est l'échelle maximale.
   Ainsi, $\forall j, k$,

   — Le coefficient $a_J[k]$ est à l'indice k dans la représentation.

   — Le coefficient $d_j[k]$ est à l'indice $\frac{N}{2^j} + k$

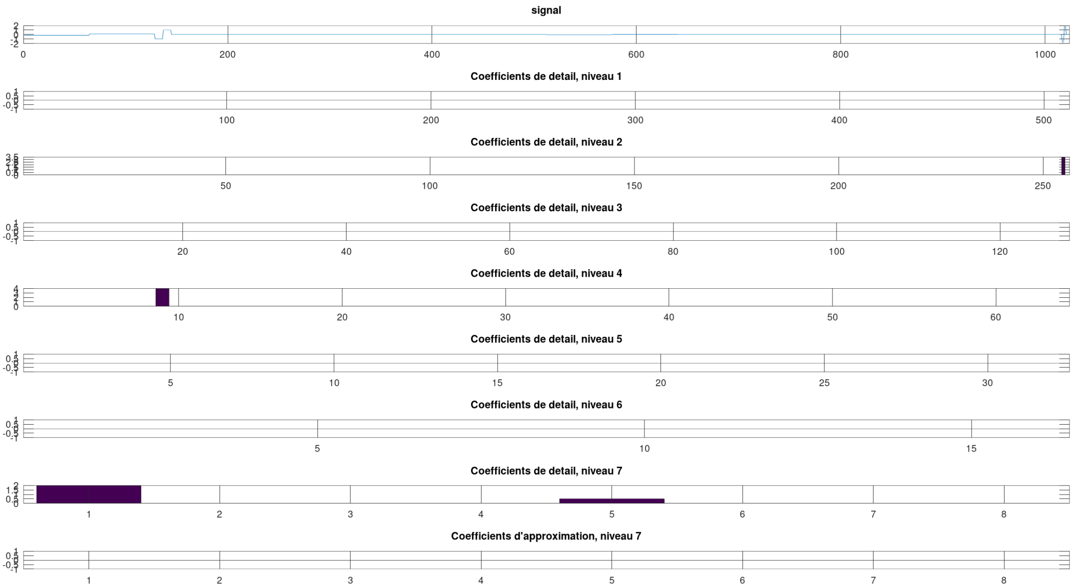
2. On souhaite construire un vecteur `DWT x` contenant seulement un coefficient non-nul de détail à la plus grande échelle et situé approximativement au milieu de l'axe temporel.
   On place alors d'après 1. ce coefficient non-nul à l'indice : $\frac{N}{2^J} + \frac{N}{2^{J+1}} = \frac{3N}{2^{J+1}}$
3.
4.
5.
6.
7.

2 Débruitage dans l'espace des ondelettes

On considèrera dans toute cette partie une ondelette de Haar.

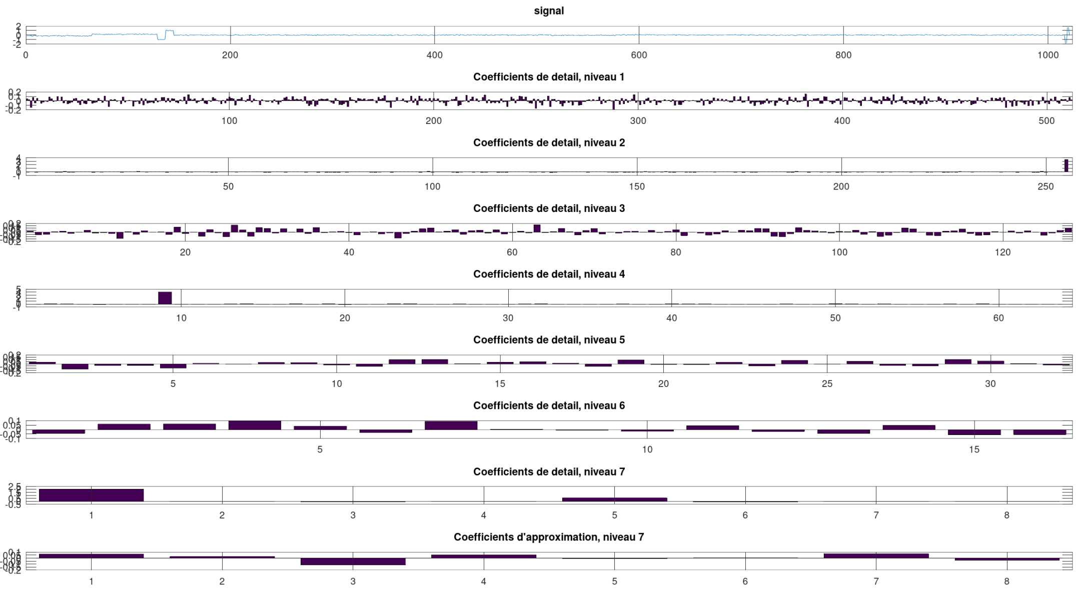
1. Créons un signal dont la DWT jusqu'à l'échelle J = 7 contient uniquement quelques coefficients de détail non nuls.
   Représentons alors le signal obtenu et sa DWT :

FIGURE 1 – Signal et sa DWT



2. Ajoutons du bruit au signal obtenu et visualisons ce nouveau signal ainsi que sa FWT :

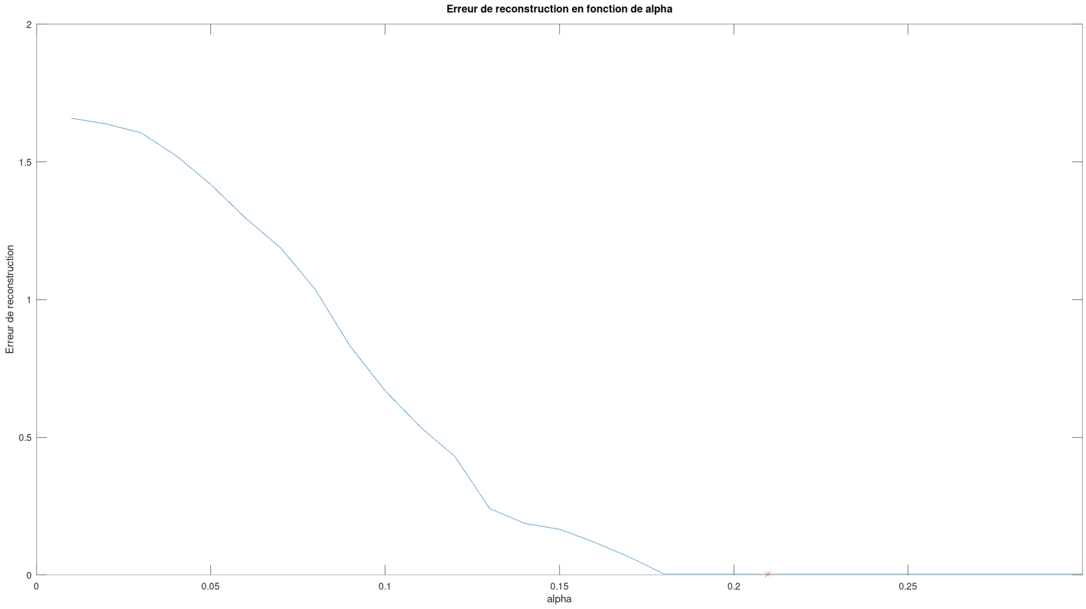
FIGURE 2 – Signal bruité et sa DWT



On observe ainsi que l'ajout de bruit au signal résulte également en un ajout de bruit à sa DWT. Toutefois on remarque que les nouveaux coefficients non nuls liés au bruit sont d'amplitude faible devant celle des coefficients de la DWT du signal sans bruit.Un filtrage des coefficients à l'aide d'un seuil devrait permettre un débruitage du signal.

3. Mettons maintenant en œuvre le processus de débruitage décrit. Représentons alors l'erreur de construction commise en fonction de la valeur du seuil α et celle correspondant au seuil $\alpha^* = \sigma_b\sqrt{2ln(N)}$:

Figure 3 – Erreur de reconstruction en fonction de alpha



4. On génère les signaux "Blocks" et "Doppler" à l'aide de la fonction `makesig`.
5. Représentons ces deux signaux ainsi que leur DWT :

Figure 4 – Singal "Blocks" et sa DWT

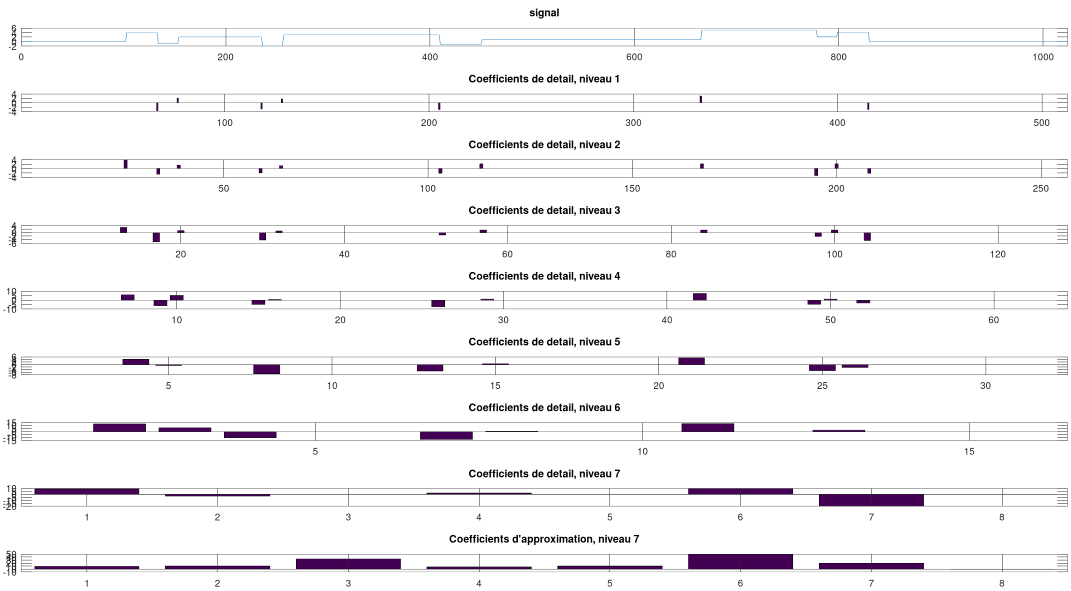
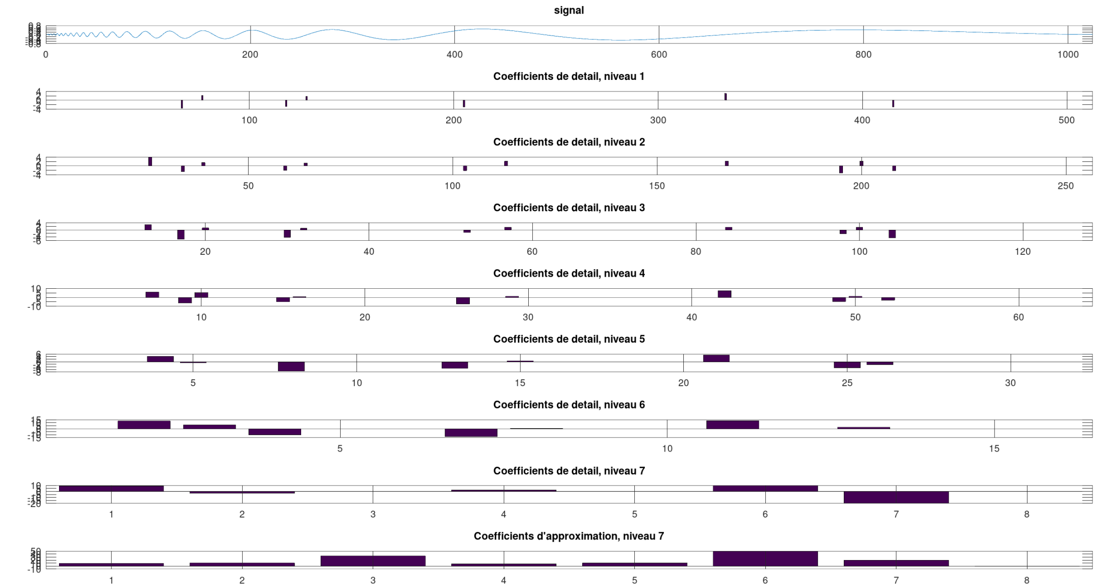


Figure 5 – Signal "Dopler" et sa DWT



Ajoutons du bruit avec un RSD de 20 dB et appliquons la procédure précédente. On obtient les tracés d'erreur de reconstruction en fonction du seuil suivants :

Figure 6 – Erreur de reconstruction en fonction du bruit (Blocks)

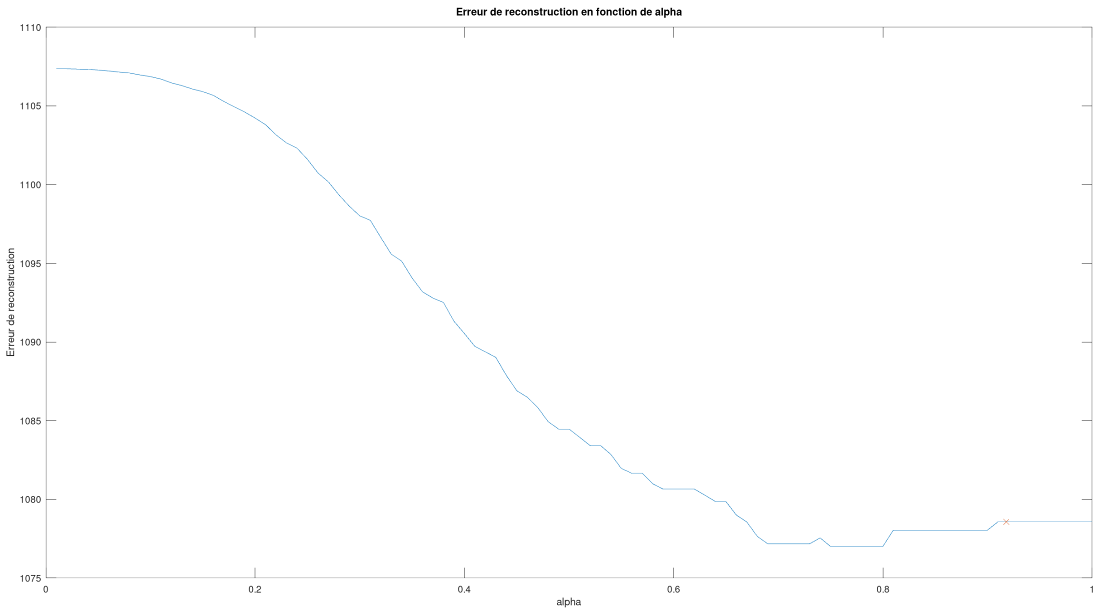
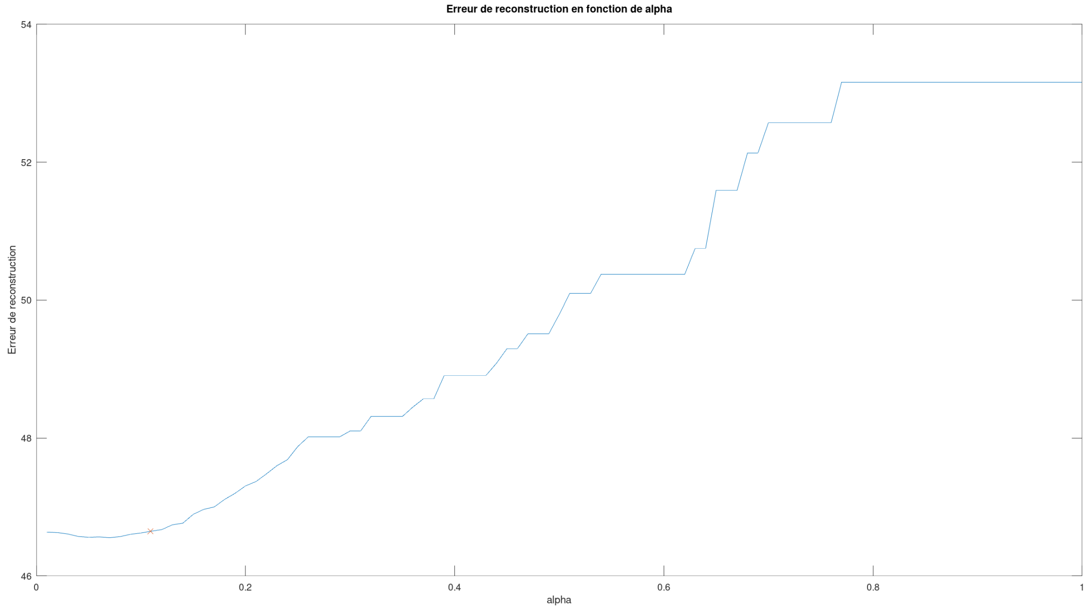


Figure 7 – Erreur de reconstruction en fonction du bruit (Dopler)



6. En utilisant maintenant une ondelette de Daubechies d'ordre 4, on obtient les tracés suivants :

Figure 8 – Erreur de reconstruction en fonction du bruit (Blocks)

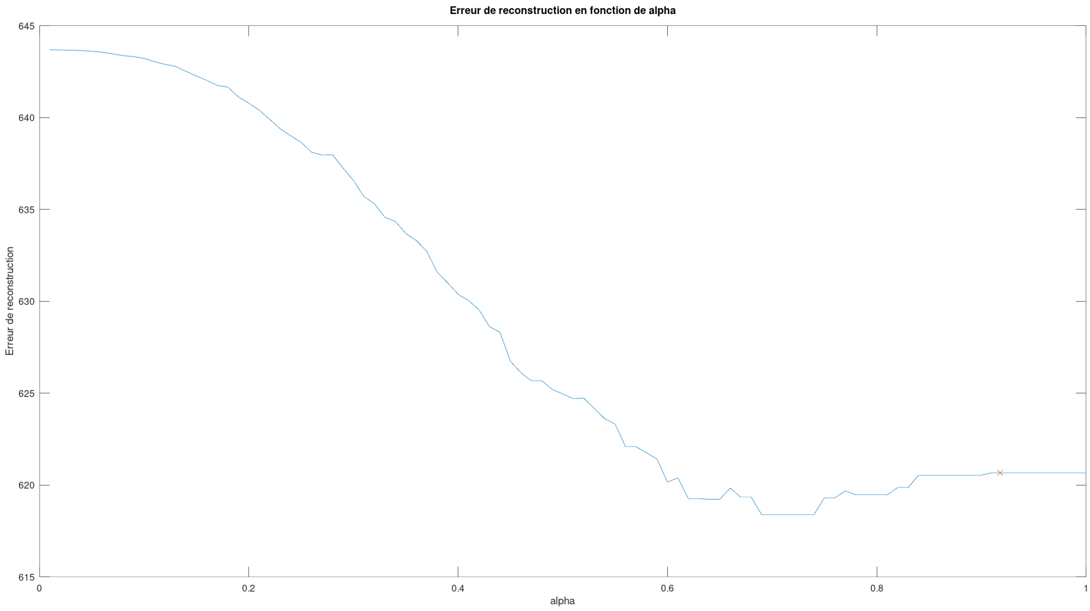
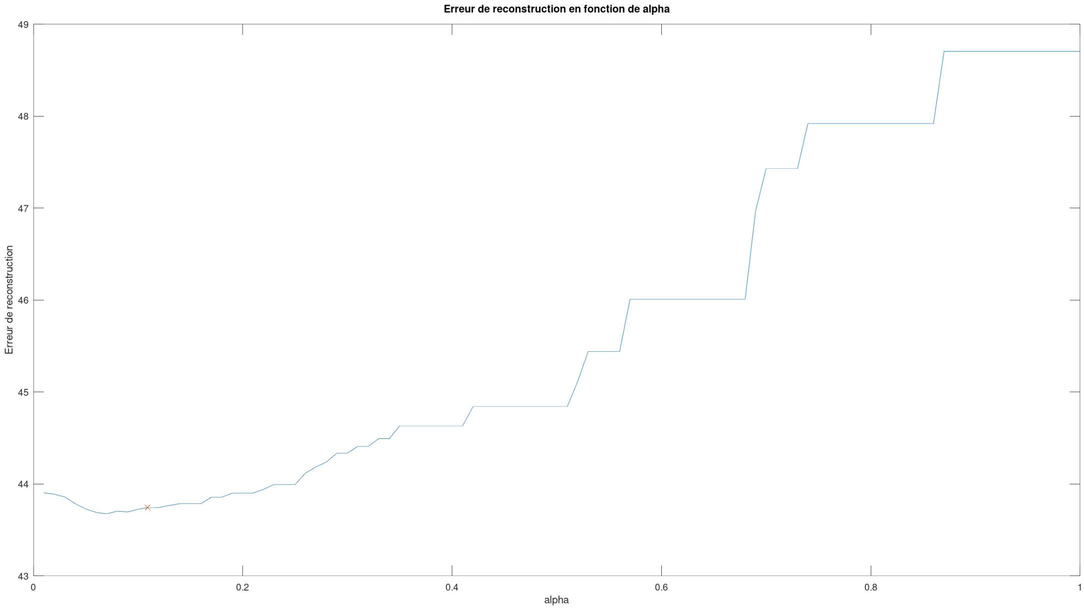


Figure 9 – Erreur de reconstruction en fonction du bruit (Dopler)



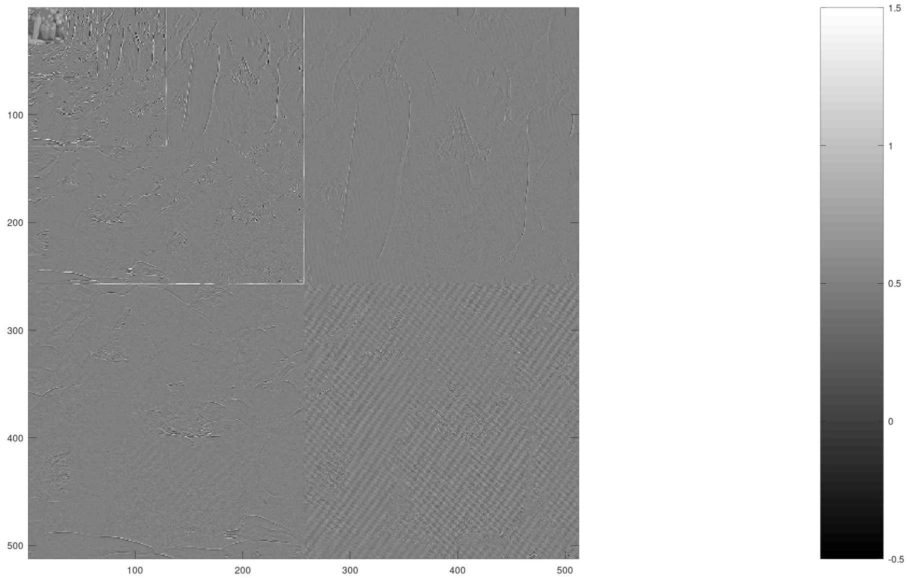
3 Compression d'images

1. On ouvre l'image `peppers.tiff` à l'aide de la fonction `imread` de Matlab.
2. Visualisions cette image :

Figure 10 – Image `peppers.tiff`

3. On calcule sa transformée en ondelettes 2D à l'échelle J = 4
4. Visualisions alors cette DWT-2D :

Figure 11 – DWT-2D



5. Représentons l'histogramme des coefficients et traçons les par ordre décroissant en valeur absolue :

Figure 12 – Histogramme

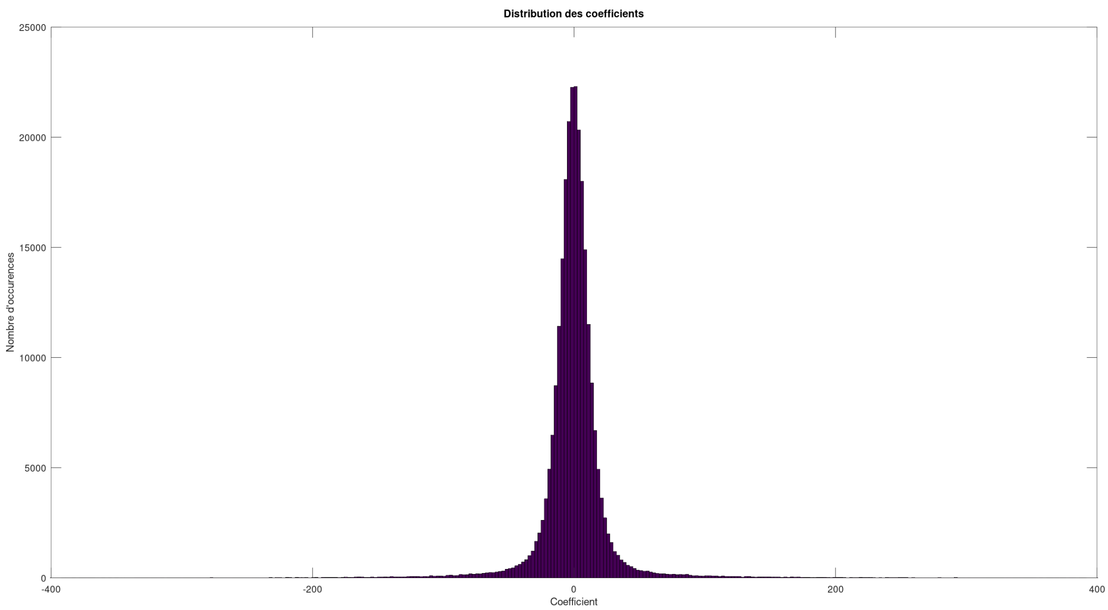
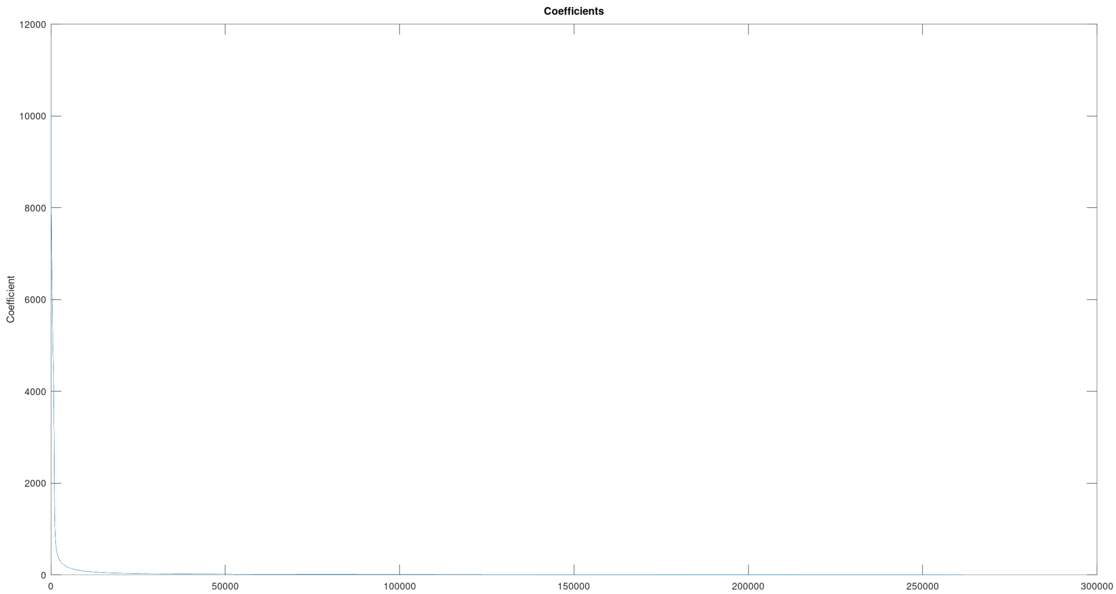


FIGURE 13 – coefficients (ordre décroissant)



6. La procédure de compression consiste en la reconstruction de l'image à partir de sa DWT seuillée. On se donne un τ et on ne conserve que les $\tau\%$ coefficients les plus grands.
7. Visualisions l'image reconstruite pour $\tau \in \{1, 5, 20\}$.

FIGURE 14 – Résultats pour $\tau = 1\%$

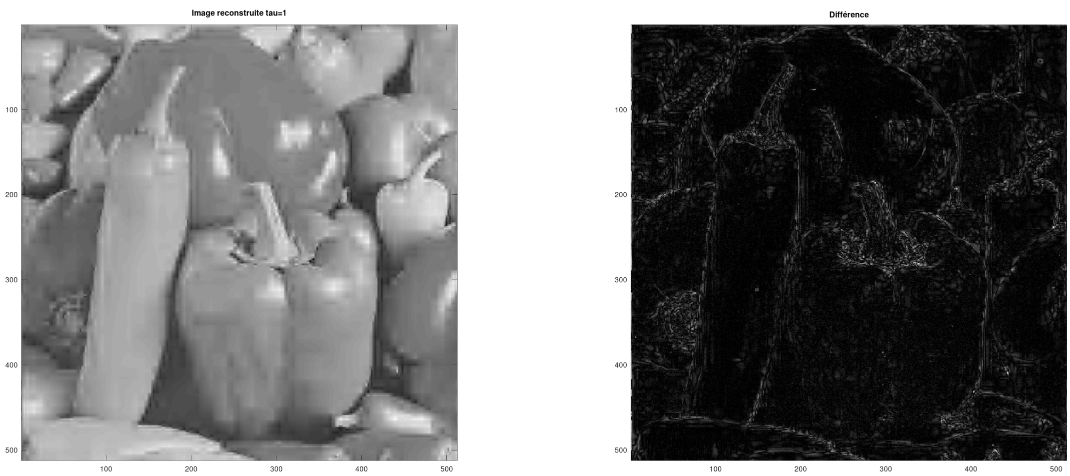


FIGURE 15 – Résultats pour $\tau = 5\%$

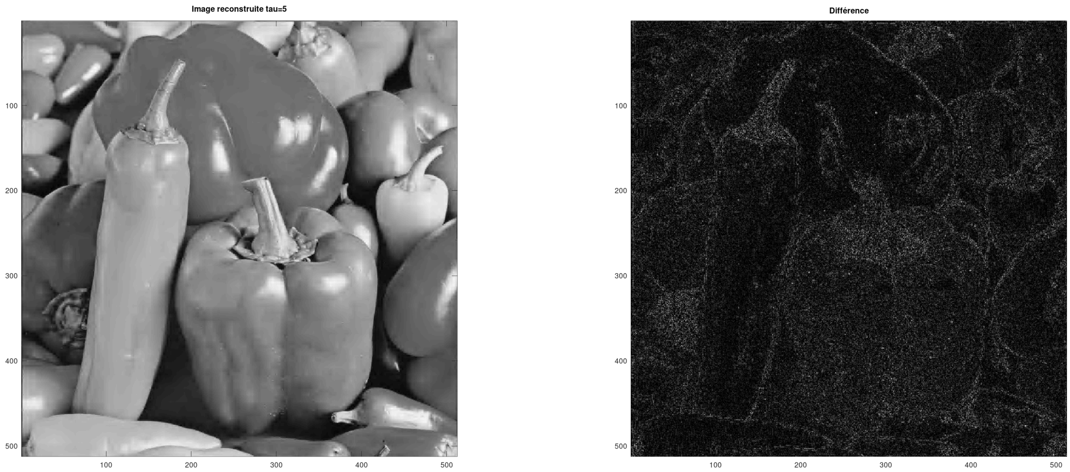
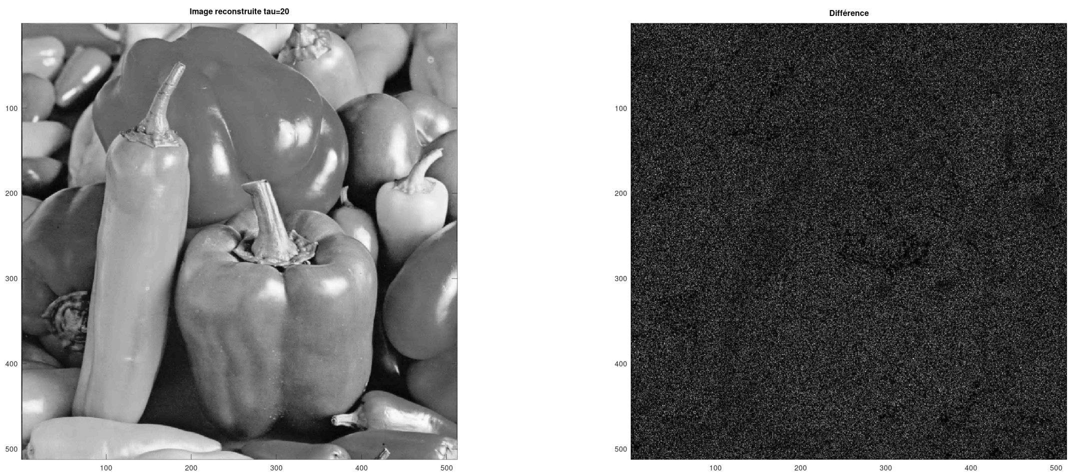


FIGURE 16 – Résultats pour $\tau = 20\%$



Conclusion