**Compte rendu de projet de Télécommunications**

Mai 2019

Théo Vazquez, Aymen Ben Abdallah,   
Hugo Georgenthum, Vincent Dambrine

Table des matières

[Présentation de la compression en format JPEG 3](#_Toc9265426)

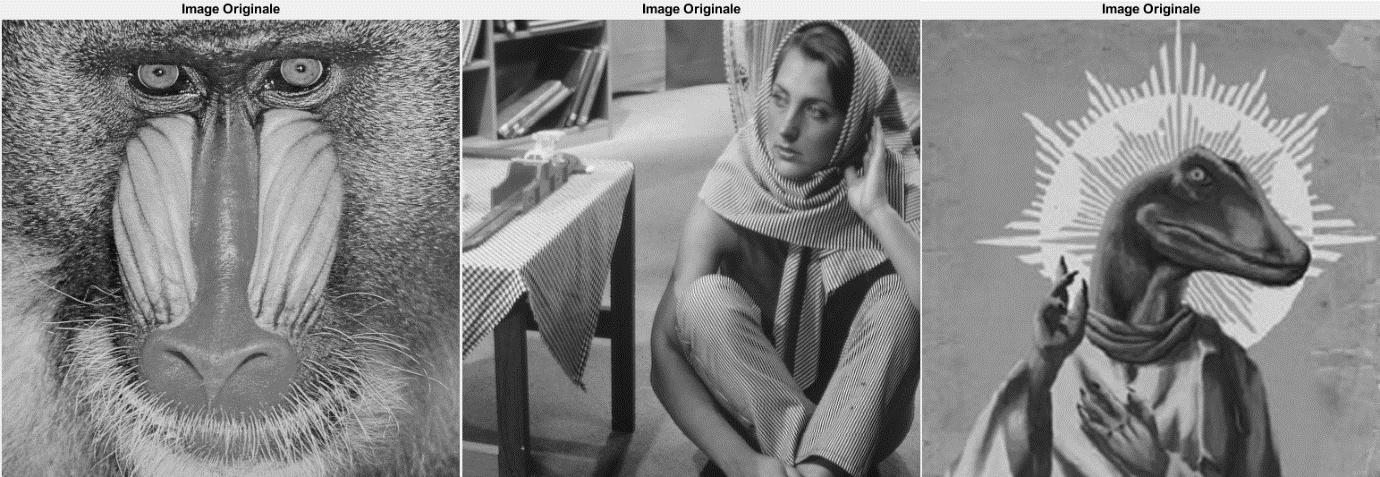
[Présentation de la décompression en format JPEG 5](#_Toc9265427)

[Observations 6](#_Toc9265428)

[Robustesse face à la transmission par canal AWGN 7](#_Toc9265429)

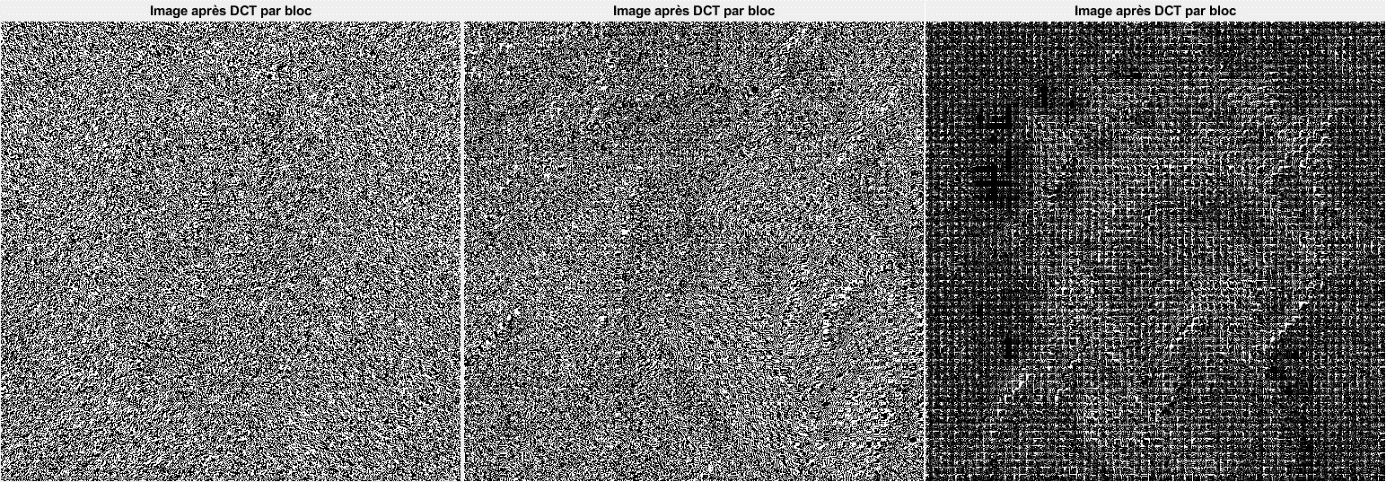
# Présentation de la compression en format JPEG

La compression en JPEG n’accepte que des images en RGB (si elle est en niveau de gris, comme barbara, il faut d’abord la convertir en RGB), carrées et de tailles multiples de 8 (par exemple, toutes nos images de test sont en 512\*512).



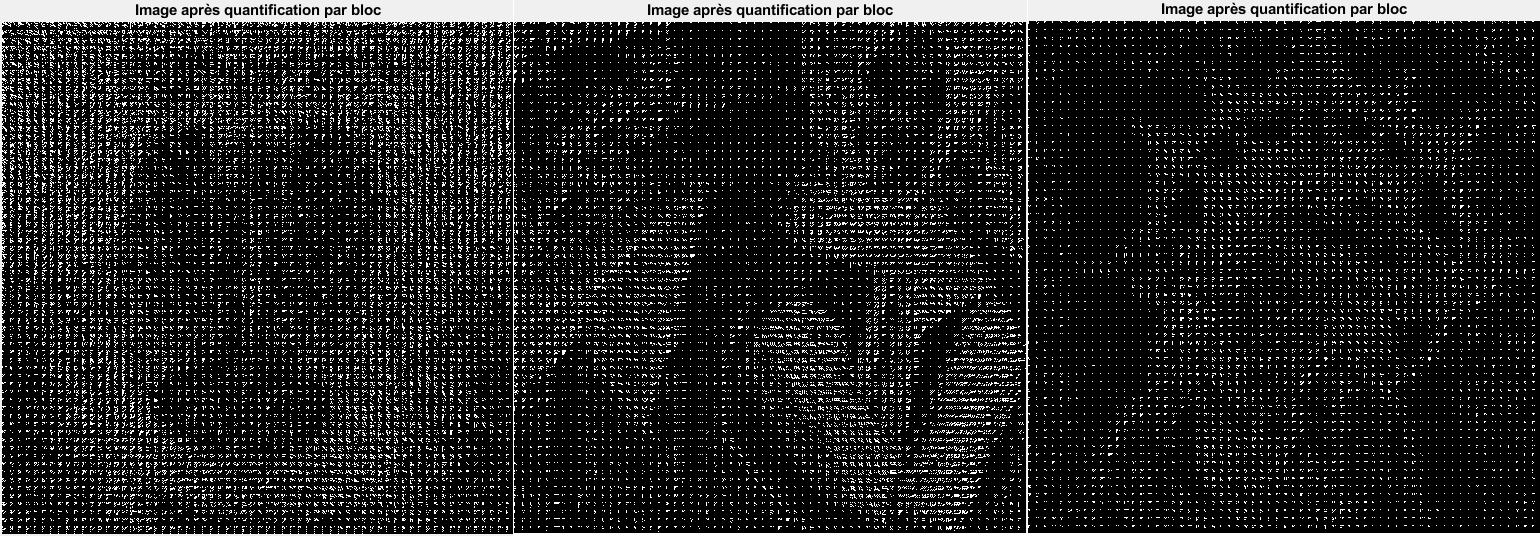
***Figure 1 :*** *Images originales*

Tout d’abord, l’image est divisée en blocs de 8\*8, puis on applique à chaque bloc la transformée en cosinus discrète (DCT).



***Figure 2 :*** *Images après application de la DCT*

Ensuite, on quantifie chaque bloc grâce à la matrice de quantification donnée (qui corresponds à la matrice de quantification JPEG pour un niveau de compression = 50%).



***Figure 3 :*** *Image après quantification*

Nous transformons alors chaque bloc en vecteur grâce à une lecture en zigzag (Un code déjà existant a été utilisé pour la lecture en zigzag).

Ces vecteurs sont ensuite codés selon le codage RLE (Le code était déjà existant), ce qui nous fournit un unique vecteur qui est lui-même codé grâce au codage de huffman (En somme, ce qui sera envoyé sera le vecteur en sortie du codage RLE, ce qui nous permet de reconstruire les symboles et les répétitions de ces symboles lors de la décompression).

Enfin, le message généré en sortie du codage de huffman est transmis par le canal de transmission.

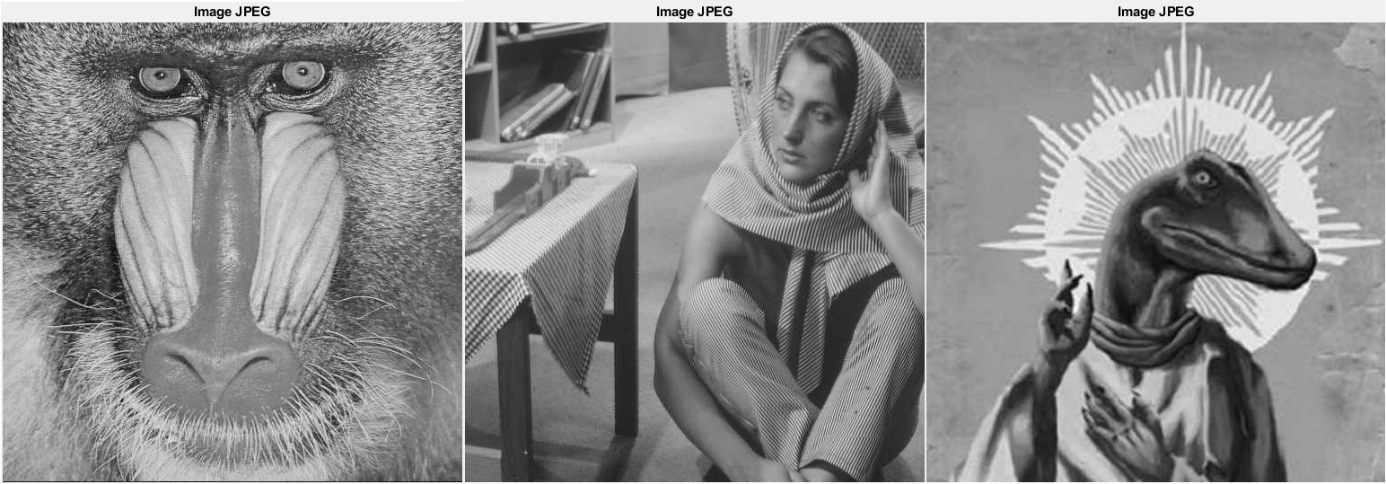
# Présentation de la décompression en format JPEG

Pour la décompression on doit recevoir deux vecteurs : le message en sortie du codage de huffman et le dictionnaire généré pour le codage de huffman. Nous n’avons pas transmis le dictionnaire via le canal pour ce projet.

D’abord nous décodons le message reçu et nous reconstruisons le vecteur en sortie de lecture en zizag par bloc. Nous reconstruisons alors chaque bloc en lecture par zizag inversée.

Nous appliquons alors l’inverse de chaque transformation (d’abord quantification inverse puis DCT inverse avant de remettre le bloc ainsi traité à sa place.

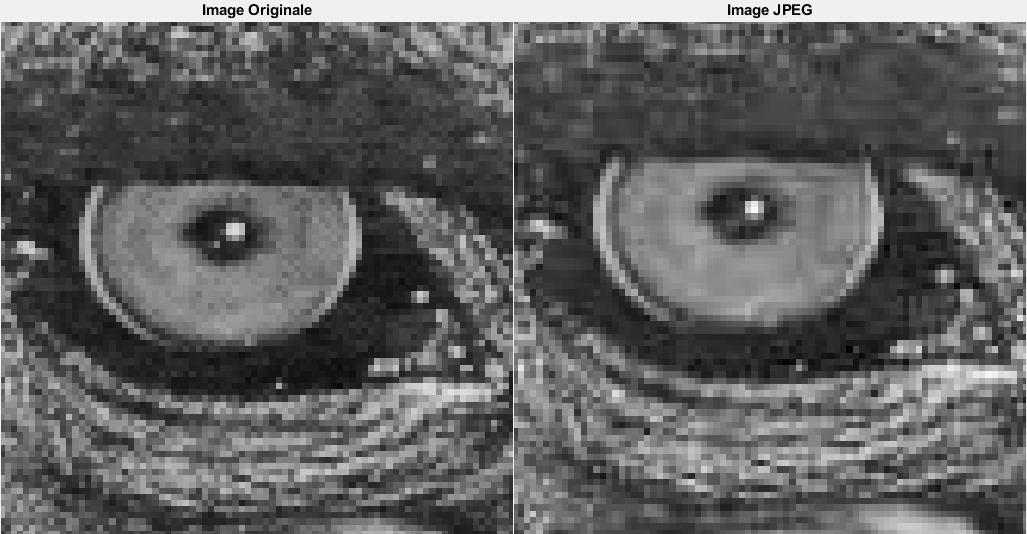
Enfin, l’image décompressée est affichée



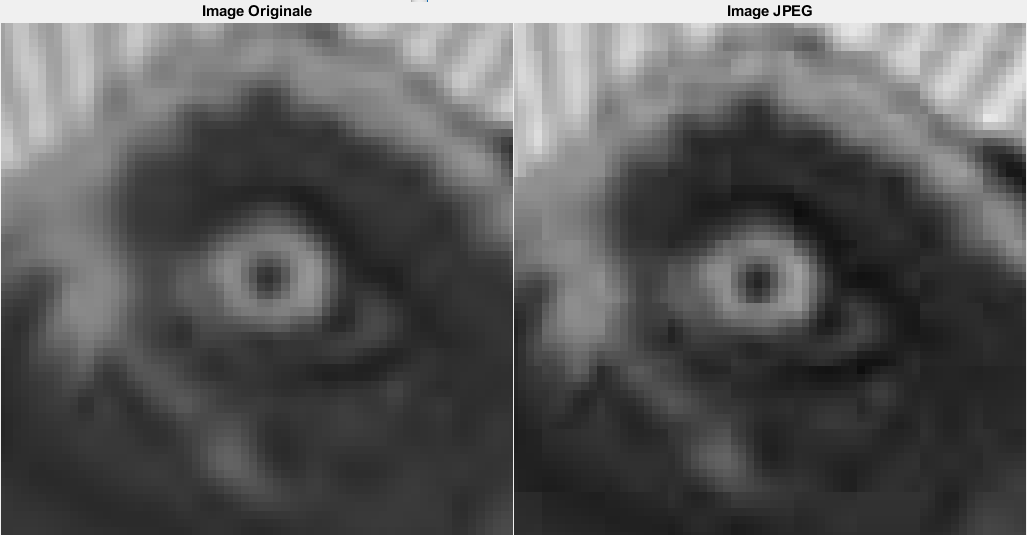
***Figure 4 :*** *Images décompressées*

# Observations

On observe dans l’image décompressée une légère perte de qualité autour des zones à détail (quadrillages surtout). De plus, on peut légèrement discerner les blocs 8\*8 :







***Figures 5 à 7 :*** *Vue de détail avant et après compression JPEG*

Néanmoins, l’image générale reste fidèle à l’image originale.

# Robustesse face à la transmission par canal AWGN

A faire