



# Compresseur universel d'images 4K

CONDE ALAZAR Arthur  
BONETTI Timothée

# Problematique

## IMAGE 4K

3840 x 2160 pixels

8,294,400 pixels pour  
une image.

Si image couleur RGB:  
24,883,200 octets soit  
24,88 Mo de memoire

Comment peut-on développer  
un compresseur universel  
capable de gérer efficacement  
les images 4K tout en préservant  
leur qualité visuelle ?

# Objectifs du projet

- **Développer un compresseur universel** : Créer 2 outil pour compresser les images 4K.
- **Préserver la qualité visuelle** : Assurer que la compression ne dégrade pas la qualité des images contrôlée avec le PSNR.
- **Optimiser les performances** : Réduire le temps de compression et de décompression.
- **Comparer les 2 methodes** : Évaluer les performances de ces deux formats dans le contexte de la compression d'images 4K et identifier leurs avantages et inconvénients.

# Etat de l'art

1992



- Quantification
- Transformée en cosinus discrète
- Codage entropique

2000



- Quantification
- Transformation en ondelettes discrète
- Codage arithmétique

2010



- Quantification adaptative
- Prédiction de blocs
- Transformée en cosinus discrète

2015



- Meilleur que JPEG
- Basé sur le codec de compression vidéo HEVC

2019



- Basé sur le codec de compression vidéo AV1

# Les solutions proposées

## Algorithme “à la JPEG”

- **Standard de compression :** Largement utilisé pour les images numériques.
- **Technologie :** Utilise la transformée en cosinus discrète (DCT).
- **Efficacité :** Très efficace pour réduire la taille des fichiers.
- **Qualité :** Peut entraîner une perte de qualité visible à des taux de compression élevés (artefacts, blocs).
- **Encodage :** Codage RLE

## Algorithme “à la JPEG2000”

- **Évolution de JPEG :** Conçu pour offrir une meilleure qualité d'image.
- **Technologie :** Utilise la transformée en ondelettes.
- **Flexibilité :** Supporte la compression avec et sans perte.
- **Qualité :** Meilleure préservation des détails et moins d'artefacts.
- **Encodage :** Codage LZ77

# Technologies utilisées



**ImGui**

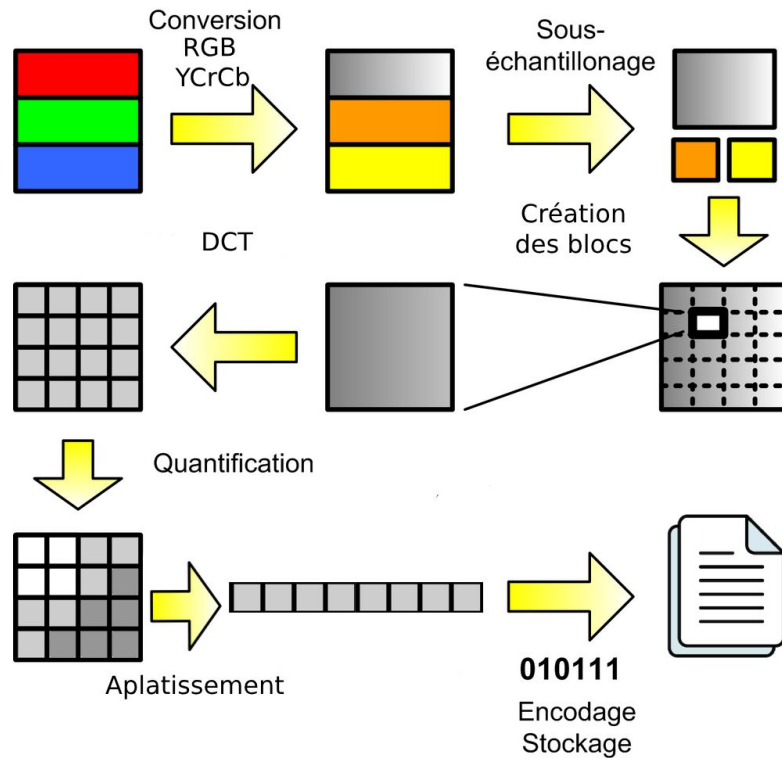
stb\_image.h

ImageBase.h



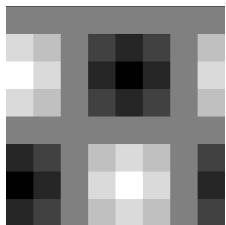
# Implémenter les méthodes proposées

# Méthode avec DCT





# DCT et quantification

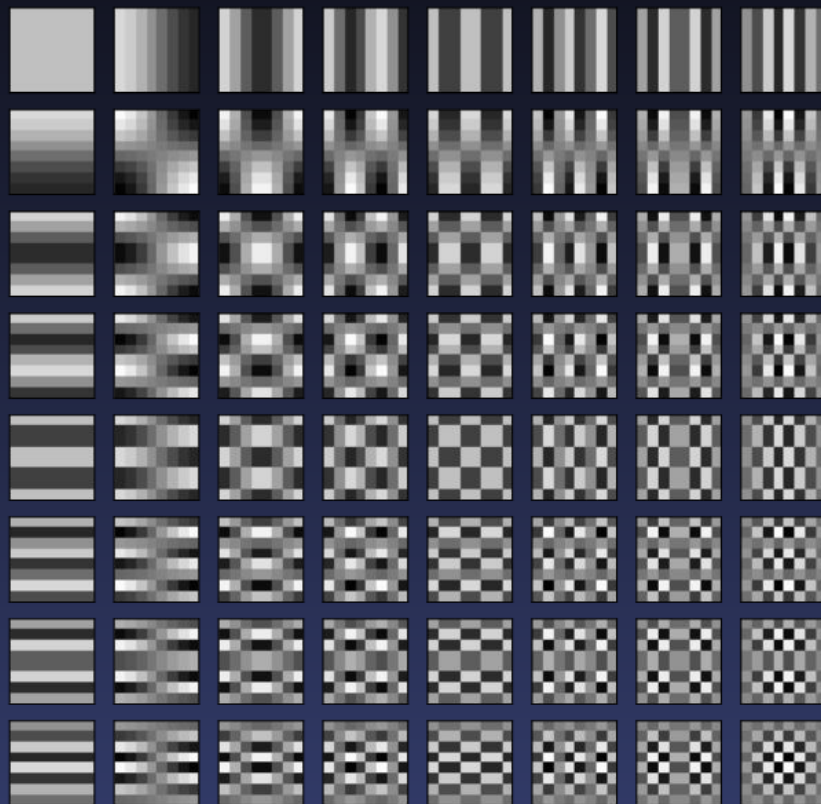


$$\begin{bmatrix} 940.00 & -54.66 & 0.00 & -5.71 & 0.00 & -1.70 & 0.00 & -0.43 \\ -36.44 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -3.81 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -1.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.29 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

Matrice de quantification

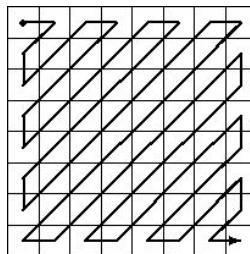
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Matrice quantifié

$$\begin{bmatrix} 59 & -5 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$


# Codage RLE

Matrice quantifiée.

$$\begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$


79	0	-2	-1	-1	-1	0	0	-1	0	0	0	0	0	0	0	0	...
(79,1)	(0,1)	(-2,1)	(-1,3)			(0,2)		(-1,1)		(0,55)							

01001000  
01100101  
01101100

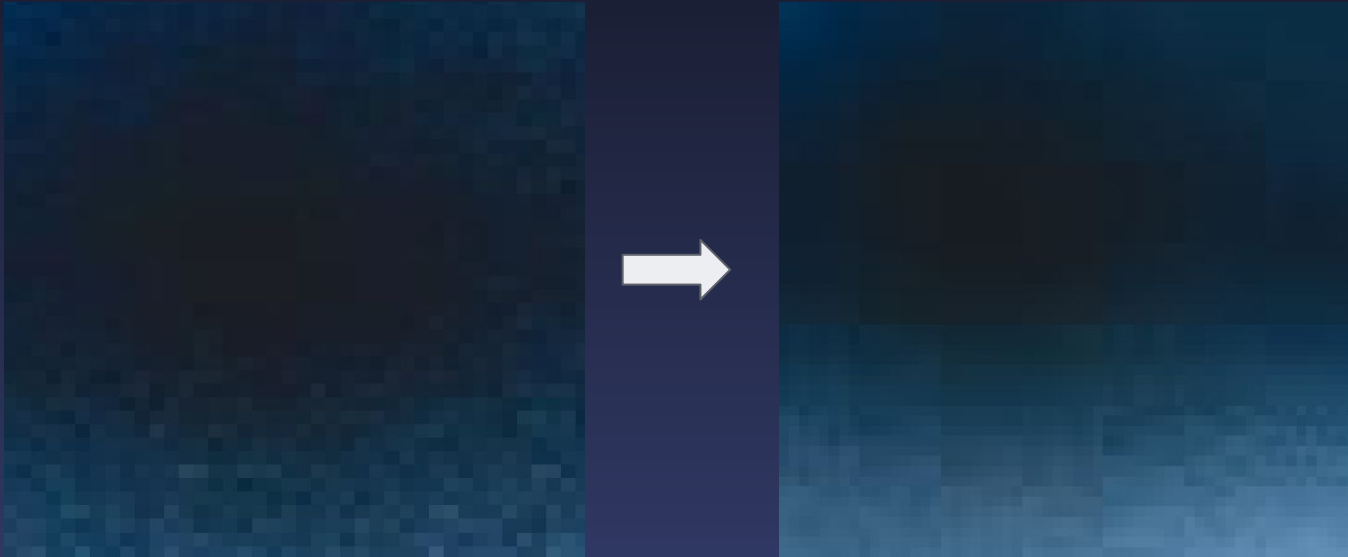


Visualisation compression DCT :  
Image 4k myrtilles



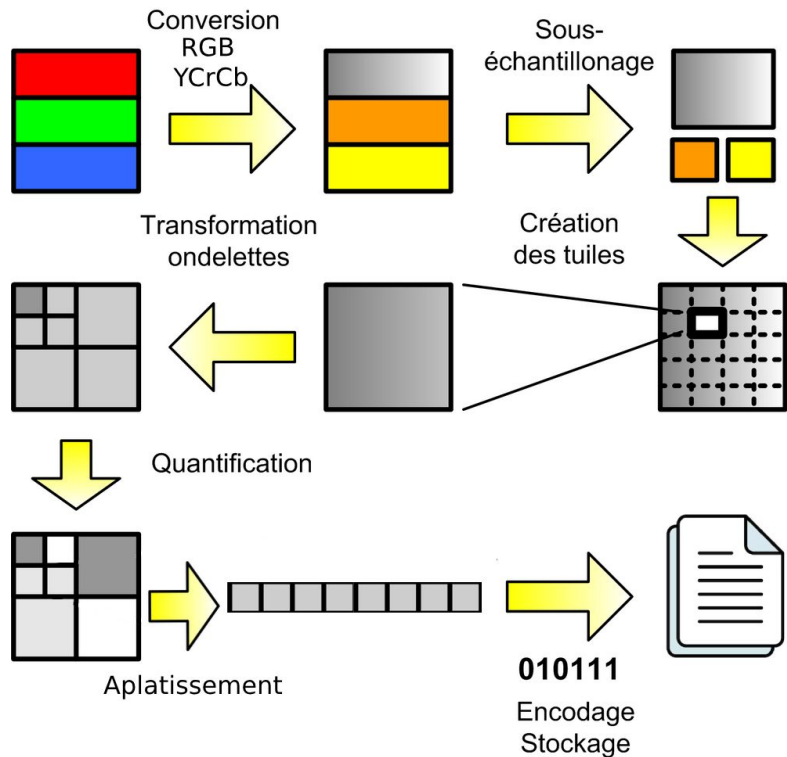
Taux de compression: 14.69  
PSNR : 32.87 dB

## Visualisation compression DCT : Image 4k myrtilles



Artefact visuel “blocking”

# Méthode avec Ondelettes



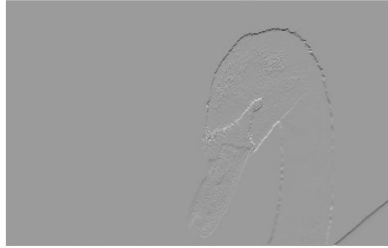
# DWT



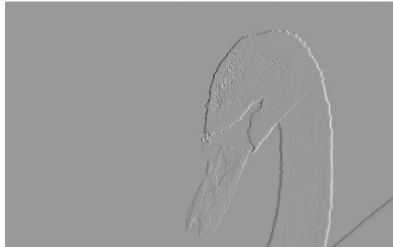
Approximation (LL)



Horizontal detail (LH)



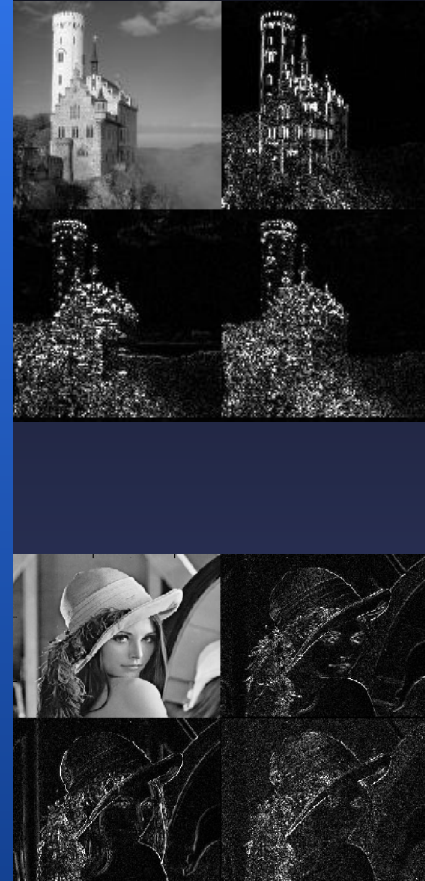
Vertical detail (HL)



Diagonal detail (HH)



Exemple Transformée en ondelettes



# Codage LZ77

Dictionnaire ou search buffer

Tampon ou look ahead buffer

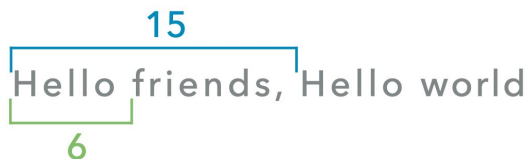


(décalage, longueur, suivant)

Fenêtre glissante ou sliding window

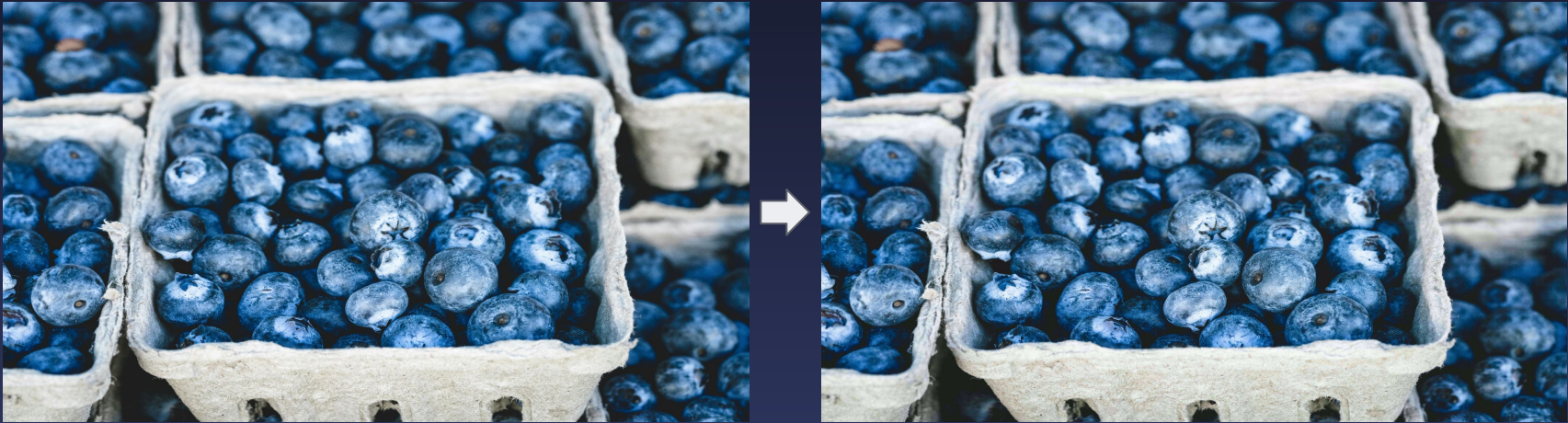
Original Hello friends, Hello world

Encoded Hello friends, (15,6)world





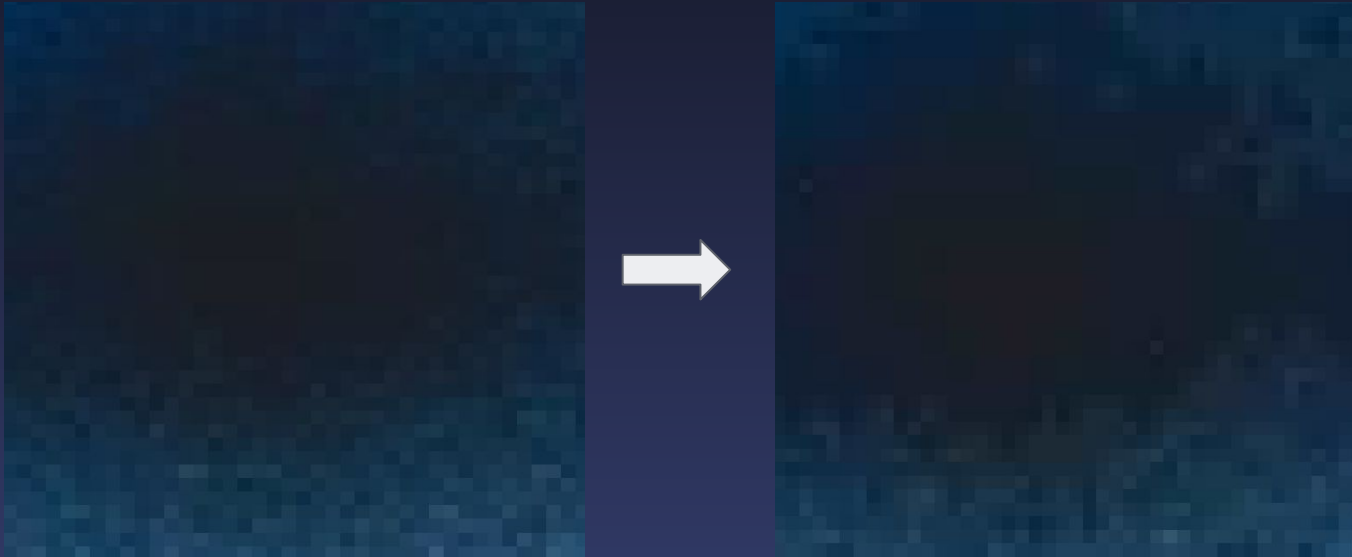
Visualisation compression ondelettes :  
Image 4k myrtilles



Taux de compression: 7.42  
PSNR : 34.17 dB



## Visualisation compression ondelettes : Image 4k myrtilles

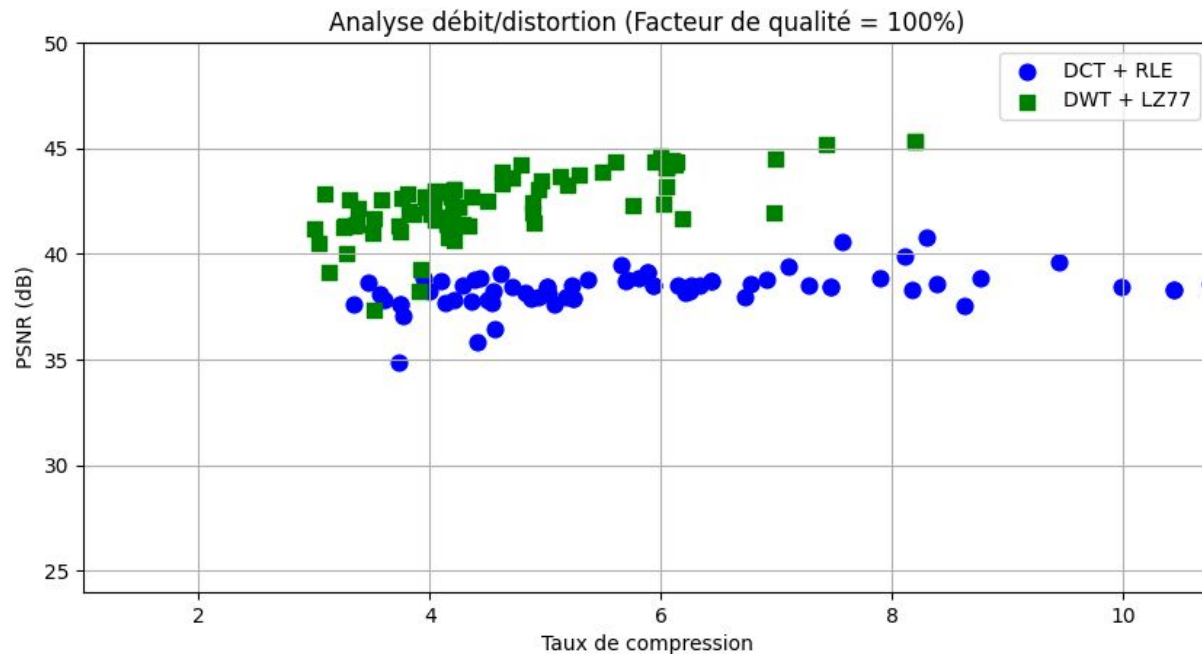


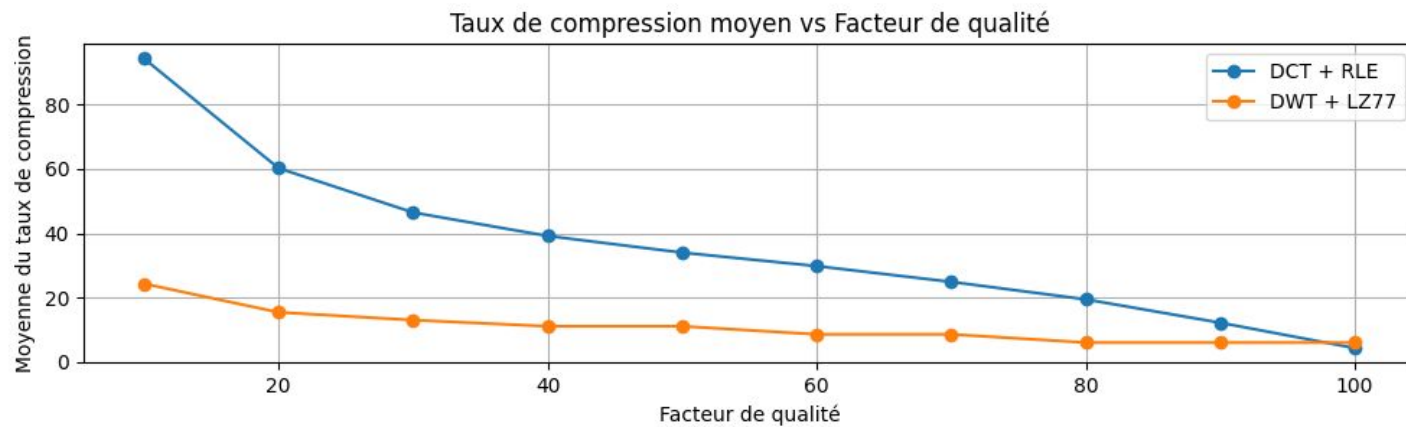
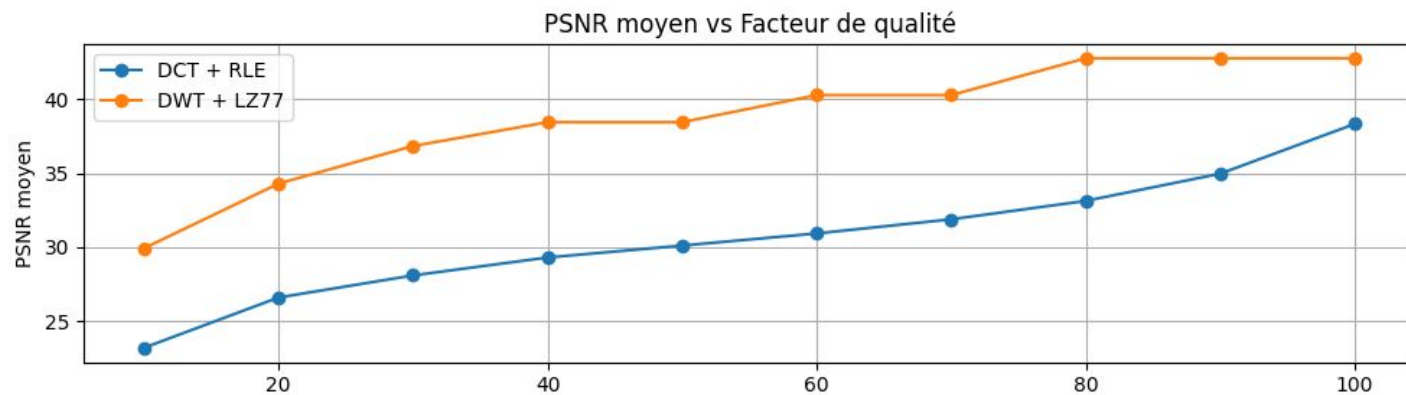
Pas de Blocking

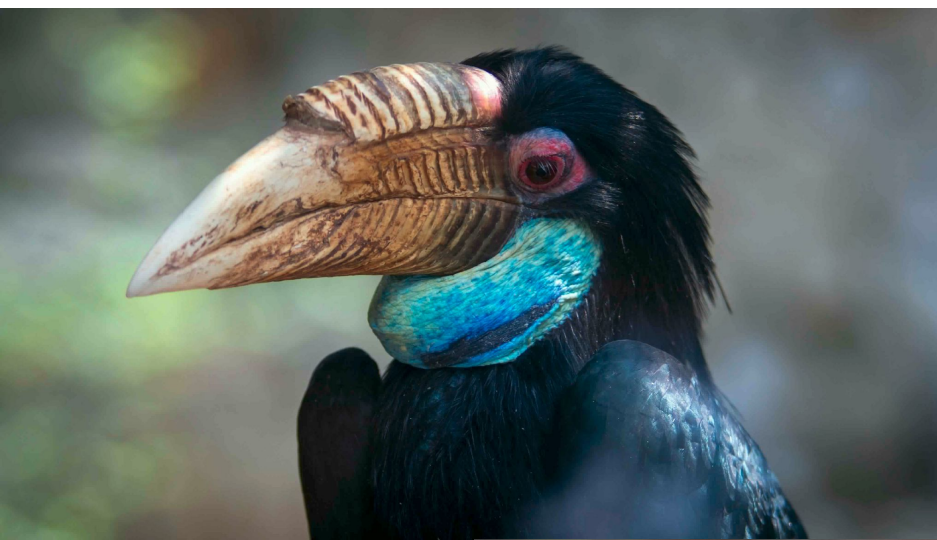


# Résultats

# Comparaison des méthodes







# Conclusion

- Première méthode orienté compression
- Deuxième méthode orienté qualité
- Des améliorations possibles