

Compte rendu 5 Projet Compression

CONDE SALAZAR Arthur et BONETTI Timothée

21 mars 2025

Table des matières

1	Amélioration de l'implémentation JPEG	2
2	Implémentation de JPEG2000	4
2.1	Quelle différence avec notre implémentation type JPEG ?	4
3	Interface	6
4	Pour la suite	7

1 Amélioration de l'implémentation JPEG

L'implémentation de la compression type JPEG étant terminée, nous nous sommes penchés sur les performances. Un des principaux problèmes était la compression et décompression des données RLE par codage de Huffman. Nous avons résolu ce problème en utilisant une hashmap qui permet d'identifier bien plus rapidement à quel code de Huffman correspond une paire RLE. Nous avons aussi ajouté du multithreading simple dans la compression et décompression. Les composantes (Y, Cb et Cr) peuvent être traitées séparément, chacune par un thread.

Ces améliorations ont grandement réduit le temps nécessaire pour la compression, mais surtout pour la décompression. Pour une image en 16 :9 au format 4K (3840 x 2160), on arrive à 6 secondes pour la compression et 6 secondes aussi pour la décompression.



FIGURE 1 – Image 4k avec en haut l'original et en bas l'image après décompression.
PSNR : 39, Taux de compression : 29.4 (par rapport au format PPM)

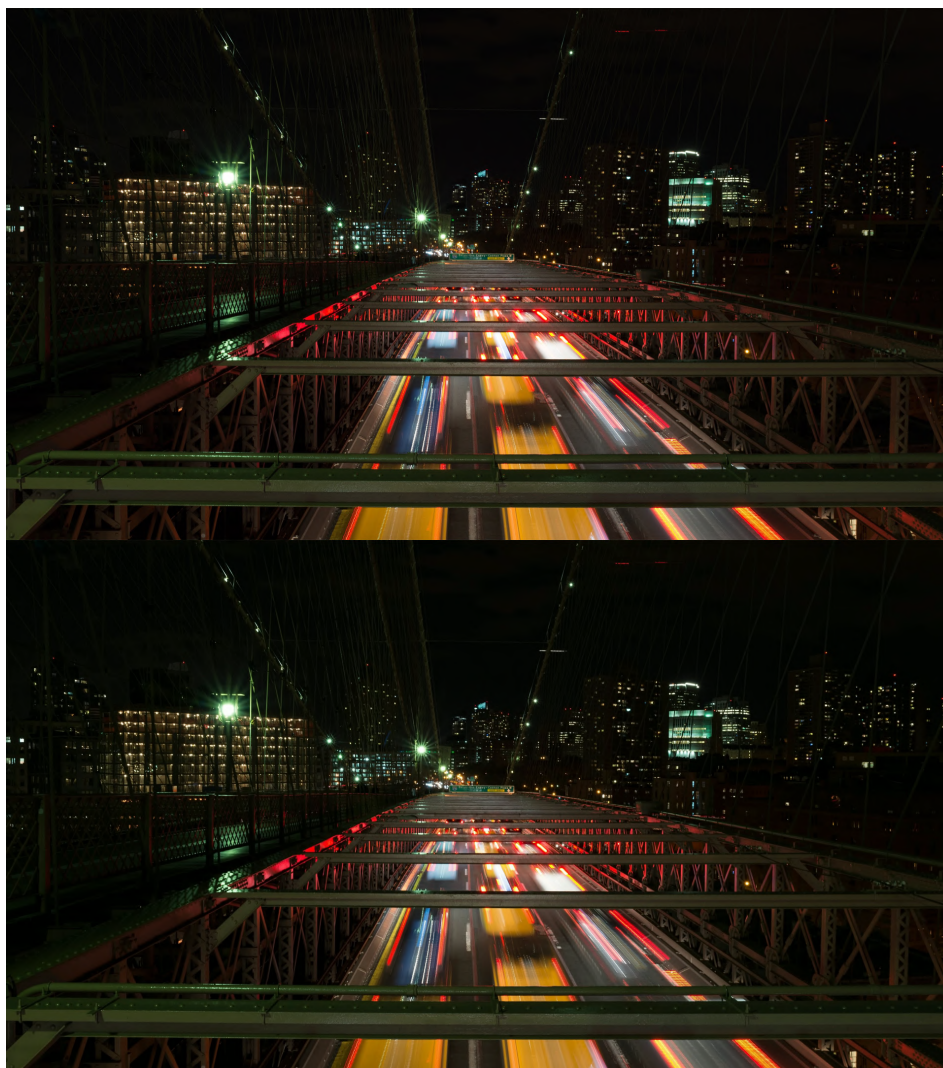


FIGURE 2 – Image 4k avec en haut l'original et en bas l'image après décompression.
PSNR : 36, Taux de compression : 21.7 (par rapport au format PPM)

2 Implémentation de JPEG2000

Nous avons bien avancé sur la compression type JPEG2000 et nous pouvons déjà obtenir des résultats. Notre implémentation suit le schéma suivant dans les grandes lignes.

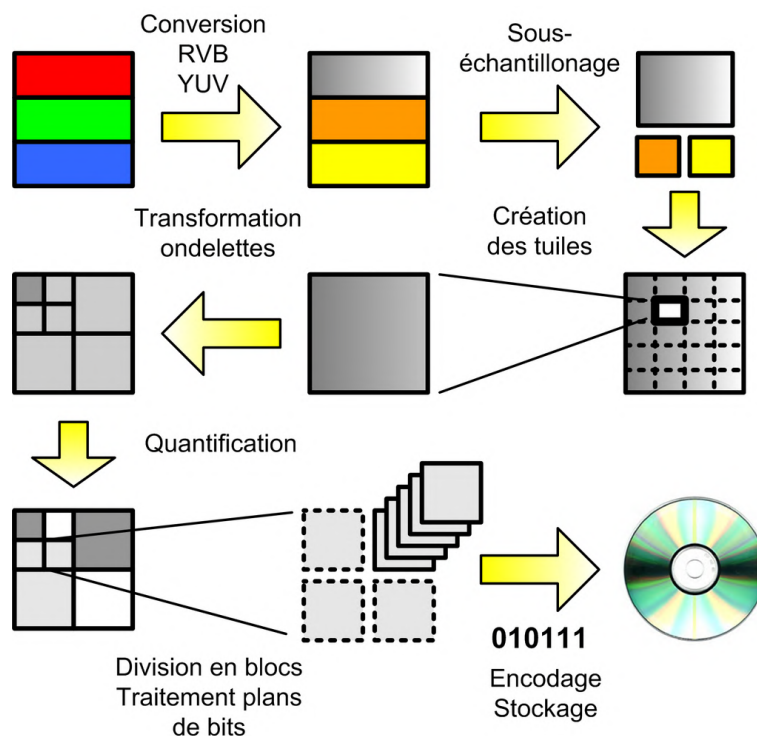


FIGURE 3

2.1 Quelle différence avec notre implémentation type JPEG ?

1. **Conversion RGB YCbCr** : Inchangée.
2. **Sous-échantillonnage** : Sous-échantillonnage bilinéaire inchangé.
3. **Création des Tuiles** : Maintenant, la taille des tuiles peut varier (pour l'instant fixée à 120 x 135).
4. **Transformation en Ondelettes** : Remplace la DCT utilisée précédemment.
5. **Quantification** : La quantification est faite de manière uniforme dans un bloc avec 2 coefficients que l'on peut faire varier pour modifier la qualité de l'image.
6. **Codage Entropique et Huffman** : Inchangé.



FIGURE 4 – Image 4k avec en haut l'original et en bas l'image après décompression.
PSNR : 40.8, Taux de compression : 4.7 (par rapport au format PPM)

3 Interface

Pour créer une interface simple, nous avons utilisé les bibliothèques suivantes :

- **SDL3** pour la gestion de la fenêtre.
- **ImGui** pour l'interface utilisateur.
- **ImGuiFileDialog** pour l'ouverture de fichiers via une boîte de dialogue.
- **stb_image** pour le chargement d'images.

Pour mieux gérer ces dépendances, nous avons passé le projet sur CMake. Il est possible de charger directement une image au format PNG, mais une copie au format PPM sera créée et utilisée pour la compression.

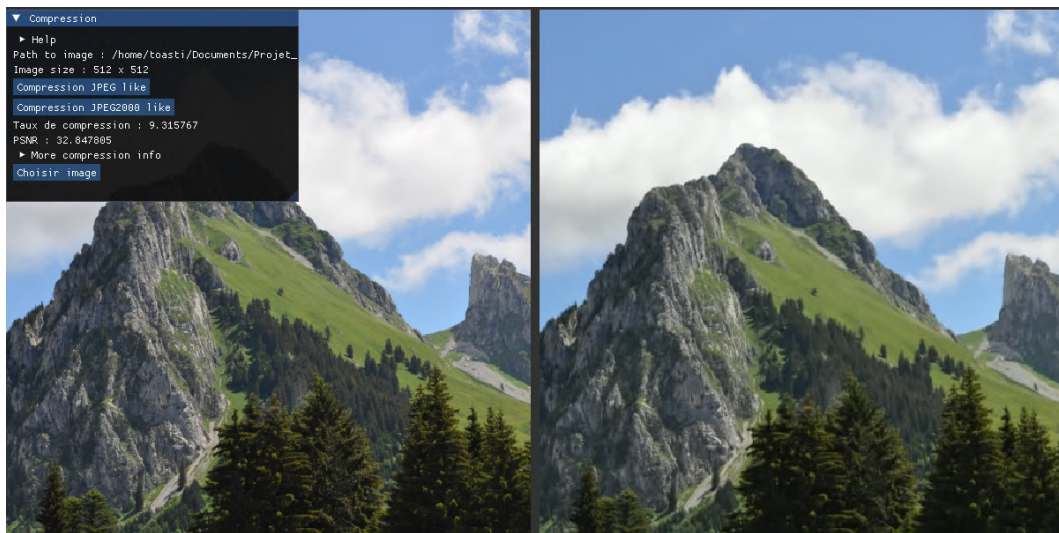
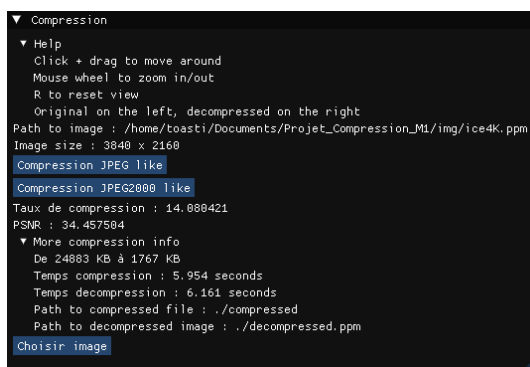
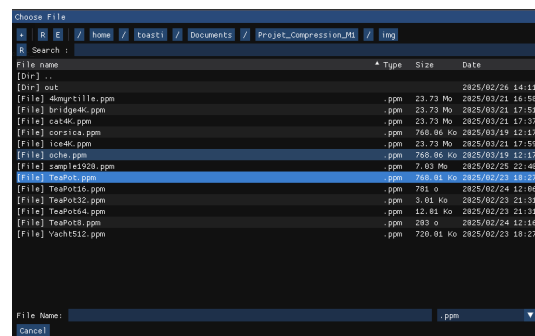


FIGURE 5



(a) Infos diverses



(b) Sélection d'une image à l'aide de ImGuiFileDialog

FIGURE 6

4 Pour la suite

Pour la suite, nous allons finir l'implémentation JPEG2000 et rendre possible de rentrer un facteur de qualité dans l'interface pour régler la qualité des compressions (pour l'instant codé en dur).