

# Résumé de l'avancé a Mi-Parcours du projet: Comparaison de deux méthodes non locales de débruitage d'images

Adam Fayret

14 octobre 2025

## 1 Introduction

Ce rapport présente l'avancement du projet d'IMA visant à implémenter et comparer deux méthodes de débruitage d'images : le *Non-Local Means* (NLM) et le *Block-Matching 3D* (BM3D). À ce stade, les implémentations des versions *pixelwise* et *patchwise* du NLM ont été réalisées et testées. Nous nous sommes mis d'accord avec notre encadrante (Elsa Angelini) lors de notre premier rdv (6 Octobre) pour faire l'implémentation de la première méthode en environ deux semaines. Puis ensuite de faire un point pour passer à l'implémentation de la seconde méthode plus complexe.

## 2 Avancées du projet

### 2.1 Mise en place des méthodes NLM

Deux versions du NLM ont été implémentées en Python :

- **NLM Pixelwise** : Chaque pixel est traité individuellement en calculant une moyenne pondérée des pixels similaires dans une fenêtre de recherche.
- **NLM Patchwise** : Les pixels sont traités par patches, ce qui réduit la redondance des calculs et améliore la qualité du débruitage.

### 2.2 Résultats préliminaires

La figure 1 présente une comparaison visuelle des résultats obtenus avec les deux méthodes NLM. Les images montrent :

- L'image originale.
- L'image bruitée (bruit gaussien).
- L'image débruitée par NLM Pixelwise.
- L'image débruitée par NLM Patchwise.

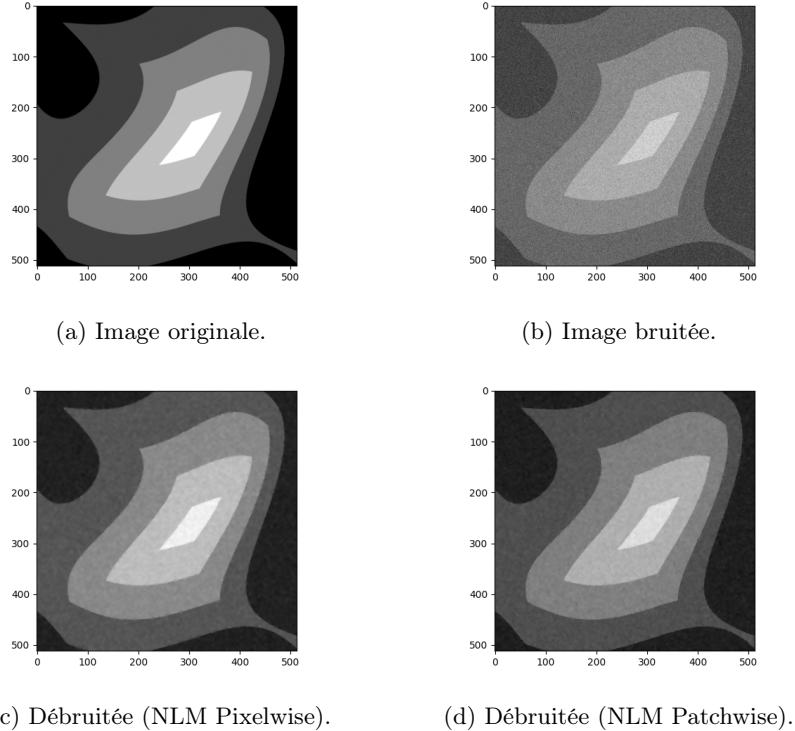


FIGURE 1 – Comparaison des résultats de débruitage.

### 2.3 Observations

- Le NLM Patchwise offre une meilleure préservation des détails et une réduction plus efficace du bruit par rapport au NLM Pixelwise.
- Les temps de calcul sont plus élevés pour le NLM Patchwise, mais cela reste acceptable pour des images de taille modérée.

## 3 Planification du travail restant

### 3.1 Amélioration de la méthode NLM

Il reste encore des améliorations à faire dans la méthode NLM. Notamment l'implémentation de padding pour éviter de rogner les patchs et les fenêtres de recherche. L'implémentation d'un filtre à posteriori peut potentiellement améliorer nos résultats.

### **3.2 Implémentation de la méthode BM3D**

La prochaine étape consiste à comprendre puis implémenter la méthode *Block-Matching 3D* (BM3D), puis de venir comparer les deux méthodes. Les tâches prévues sont :

- Étudier l'algorithme BM3D et ses étapes clés (recherche de blocs similaires, regroupement 3D, filtrage collaboratif).
- Implémenter une version simplifiée en Python, en utilisant des bibliothèques comme NumPy pour optimiser les calculs.
- Valider l'implémentation avec des tests sur des images standards.

### **3.3 Bonus**

Au vu du temps de calcul nécessaire pour ces méthodes de débruitage, si les temps deviennent trop élevés nous allons voir pour venir implémenter des tâches en parallèle ou du multi-processing (via Threads).

## **4 Conclusion**

Nous sommes légèrement en avance sur l'implémentation des méthodes de NLM. Nous allons donc voir pour venir implémenter la méthode de BM3D puis voir pour des optimisations possibles sur les deux méthodes.