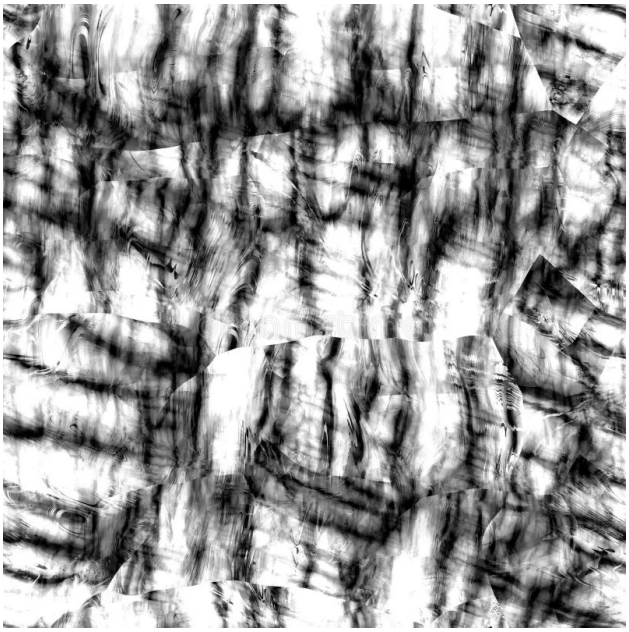
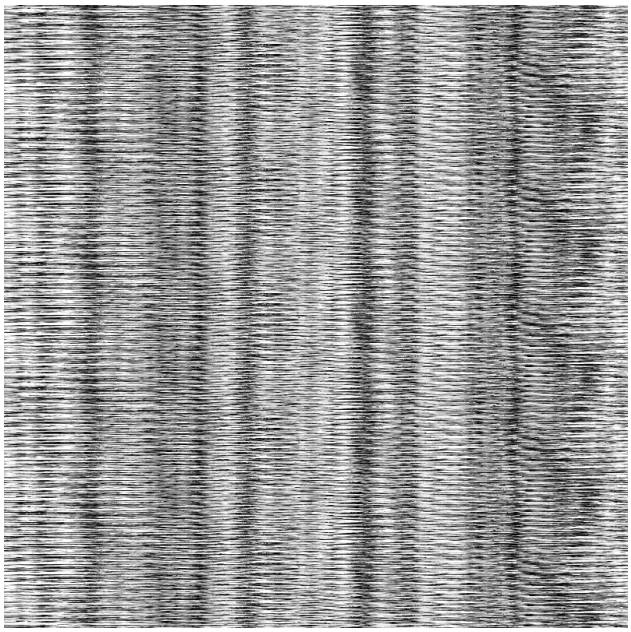
Comparaison des différents critères et méthodes de déchiffrement d'une image brouillée

Par Jules Jeanrot et Valentin Beuret
Le 6 Janvier 2026





Informatique
iut Nord Franche-Comté

Table des matières

I. Introduction

L'enjeu principal de cette étude est de déterminer l'efficience de ces deux approches selon trois axes complémentaires : la précision des résultats, la complexité algorithmique théorique et la vitesse d'exécution réelle. Dans cette optique, cinq critères de calcul ont été implémentés et analysés afin d'évaluer la similarité entre les images déchiffrées et l'image de référence : la distance euclidienne, le coefficient de corrélation de Pearson, la distance de Manhattan, la corrélation croisée normalisée et la divergence de Kullback-Leibler.

Le rapport est structuré en trois parties distinctes.
Dans un premier temps, nous présentons le principe du Brouillimg, puis la mise en place des protocoles de mesure des performances à l'aide de notre programme chronomètre. La seconde partie est consacrée à l'analyse comparative des différents critères, avec une attention particulière portée aux performances de breakKey1 (BK1) et de breakKey2 (BK2). Enfin, la dernière partie propose un bilan synthétique des résultats obtenus, permettant d'identifier l'approche la plus pertinente.

II. Mise en place

Tous les tests ont été effectués sur un ordinateur portable ASUS ROG Strix de 2022 possédant une AMD Ryzen 7 4800h ainsi que 16 GB de mémoire vive, ou sur un modèle similaire avec un processeur Ryzen 5.

Les tests ont été effectués sur des images similaires de hauteur 1000px, 2000px, 3000px ou 5000px, et de largeur de 1000px.

A) Brouillimg

Nous avons deux méthodes pour casser la clé, sobrement appelées breakKey et breakKey2. La méthode breakKey utilise un principe de brute-force et est généralement très précis, la méthode breakKey2 trouve mathématiquement la clé, et est environ 1000 fois plus rapide. Elle est toutefois un peu moins précise (pour une image classique fluide, BK2 trouve la bonne clé environ 95% du temps contre virtuellement 100% pour BK1).

B) Chronomètre

Les mesures de temps d'exécution ont été réalisées à l'aide du fichier ChronometrePro.java. Ce dernier intègre également une fonctionnalité permettant d'évaluer la fiabilité des méthodes, en fournissant un pourcentage de réussite basé sur les résultats obtenus.

III. Analyse

L'analyse est répartie en cinq parties, chacune correspondant à un critère de similarité.
Les résultats obtenus pour chaque méthode de BreakKey1 (BK1) ont été mesurés 15 fois, ce qui implique une précision statistique plus limitée, et pour le critère de Kullback-Leibler cela a était testé sur seulement des images de 1000px et 2000px. En revanche, les résultats pour chaque méthode de BreakKey2 (BK2) ont été évalués sur environ 1 000 itérations, offrant ainsi une meilleure fiabilité des mesures.

A) Euclide

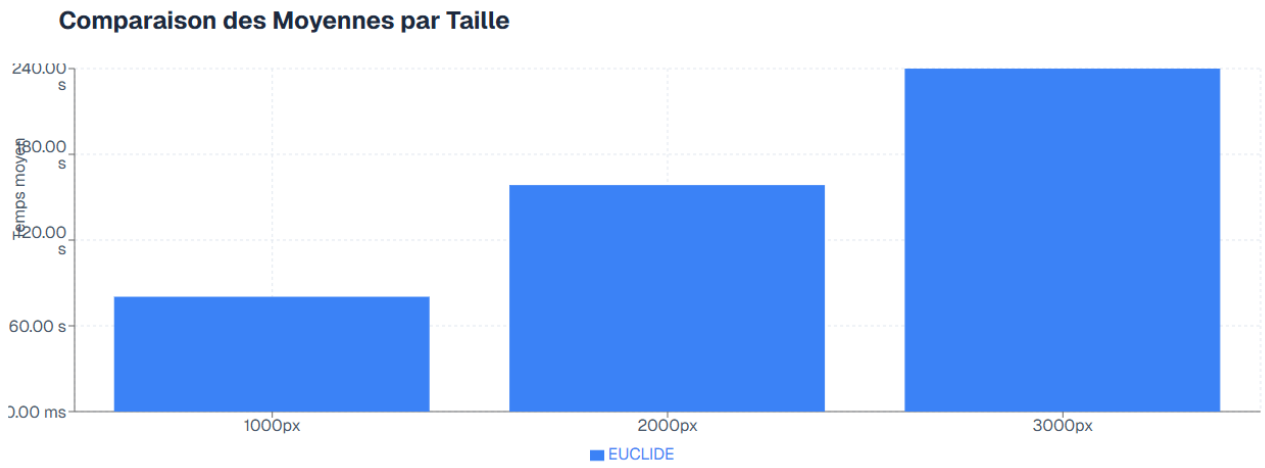


Figure 1: Comparaison de la méthode Euclide avec BreakKey1 sur différente longueur d'image

Le premier graphique révèle une progression linéaire du temps moyen d'exécution en fonction de la longueur de l'image (en pixels). L'unité de mesure sur l'axe des ordonnées est la seconde (s), montrant que le traitement de 1000px nécessite une moyenne de 80,34 s, montant jusqu'à environ 240 s pour 3000px. Cette latence importante suggère que l'implémentation BK1 souffre d'un manque d'optimisation algorithmique ou d'une exécution purement séquentielle, rendant le calcul du critère d'Euclide extrêmement coûteux dès que la taille de l'entrée augmente.

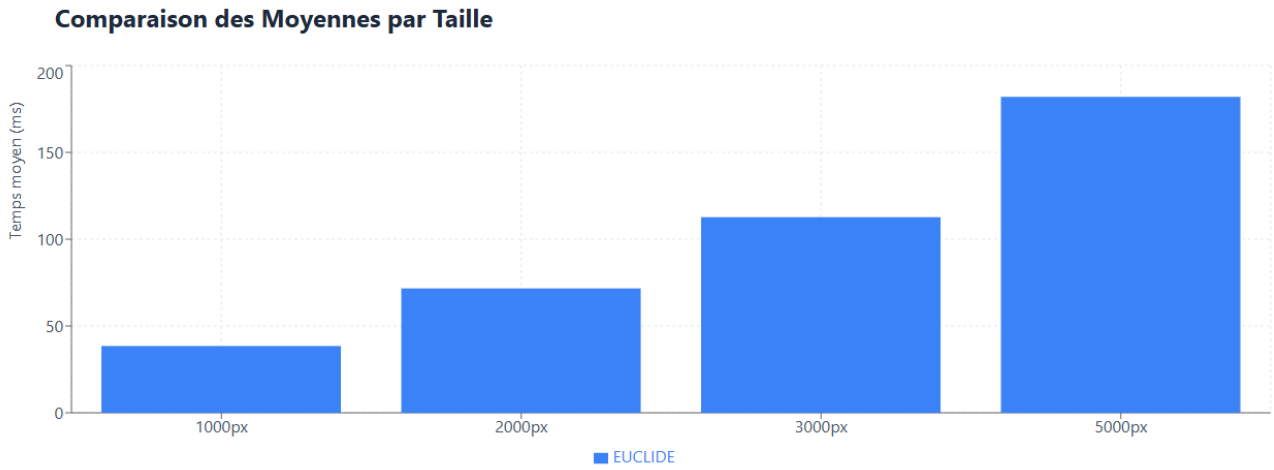


Figure 2: Comparaison de la méthode Euclide avec BreakKey2 sur différente longueur d'image

Le second graphique présente une rupture technologique majeure, caractérisée par un changement d'échelle radical : l'unité de mesure sur l'axe des ordonnées passe de la seconde à la milliseconde (ms). Bien que la tendance reste linéaire, confirmant une complexité algorithmique théorique identique (O(n)), l'efficacité est décuplée. Le traitement de 1000px ne nécessite plus que 41,40 ms en moyenne, et celui de 3000px environ 124,39 ms. Cette réduction du temps d'exécution suggère que l'implémentation BK2 bénéficie d'optimisations avancées. On note toutefois, d'après les données statistiques, une variabilité plus importante (écarts-types élevés) que sur la première configuration, indiquant une sensibilité accrue aux conditions d'exécution malgré une vitesse globalement supérieure.

B) Pearson

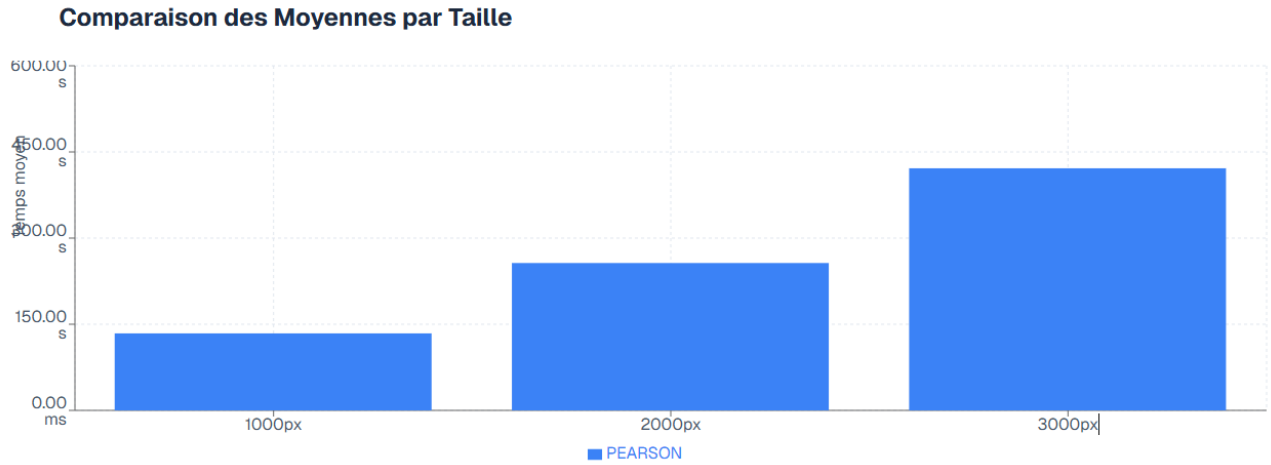


Figure 3: Comparaison de la méthode Pearson avec BreakKey1 sur différente longueur d'image

Le graphique relatif à la configuration BK1 pour le critère de Pearson montre une progression linéaire du temps de calcul extrêmement lente. L'unité sur l'axe des ordonnées est la seconde (s). Pour une image de 1000px, le temps moyen avoisine les 130s, et il grimpe drastiquement pour atteindre environ 420s (soit 7 minutes) pour une taille de 3000px. Le coefficient de Pearson étant une mesure statistique impliquant des calculs de covariance et d'écarts-types, cette latence confirme que l'implémentation BK1 n'utilise aucune optimisation de calcul parallèle, rendant le traitement de gros volumes de données quasiment inexploitable en temps réel.

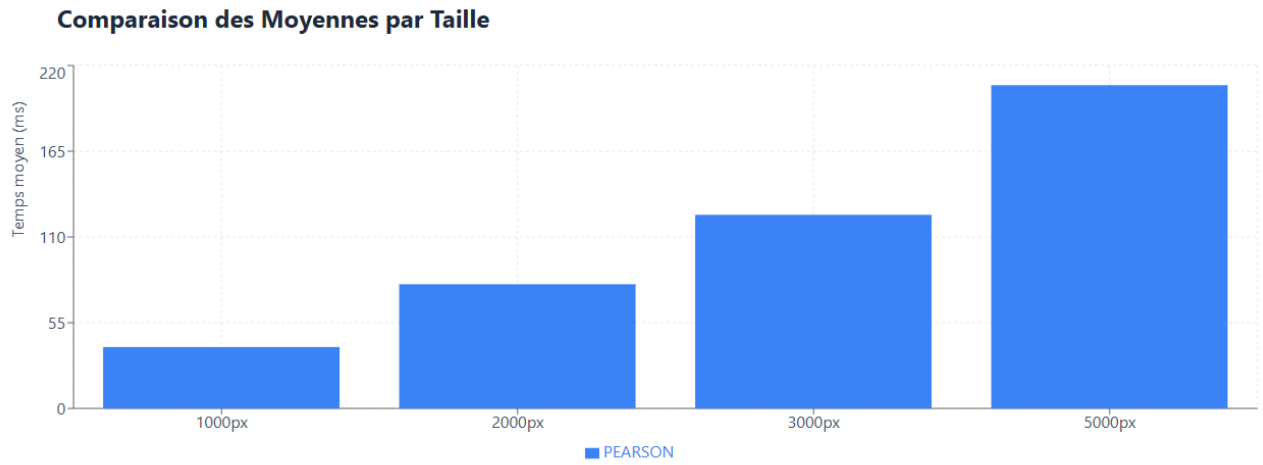


Figure 4: Comparaison de la méthode Pearson avec BreakKey2 sur différente longueur d'image

À l'inverse, le graphique de la configuration BK2 pour Pearson affiche une efficacité radicalement supérieure, avec un axe des ordonnées exprimé en millisecondes (ms). La croissance reste linéaire, mais les ordres de grandeur n'ont plus aucune mesure commune, le traitement de 1000px s'effectue en environ 40ms, tandis que celui de 3000px se stabilise autour de 125ms. Même pour une charge de travail étendue à 5000px, le temps reste inférieur à 220ms

C) Manhattan

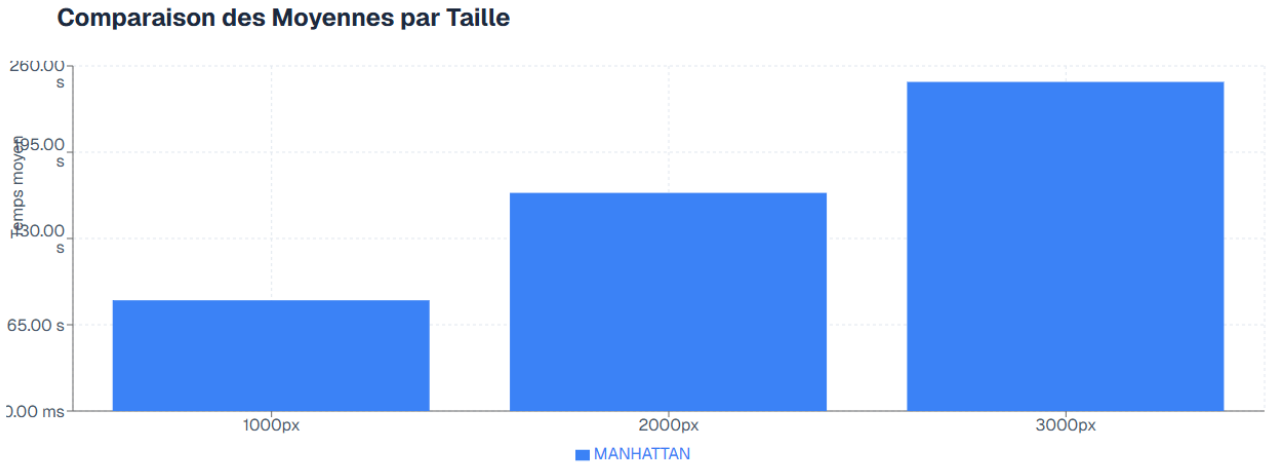


Figure 5: Comparaison de la méthode Manhattan avec BreakKey1 sur différente longueur d'image

Le graphique de la configuration BK1 pour le critère de Manhattan illustre une progression strictement linéaire du temps moyen de traitement en fonction de la résolution de l'image. L'unité de mesure sur l'axe des ordonnées est la seconde (s). On observe que pour une entrée de 1000px, le temps d'exécution est d'environ 85s, et il s'élève à près de 250s pour 3000px. Bien que la distance de Manhattan soit mathématiquement plus simple que celle d'Euclide (évitant les carrés et racines carrées), cette latence reste extrêmement élevée dans cette configuration.

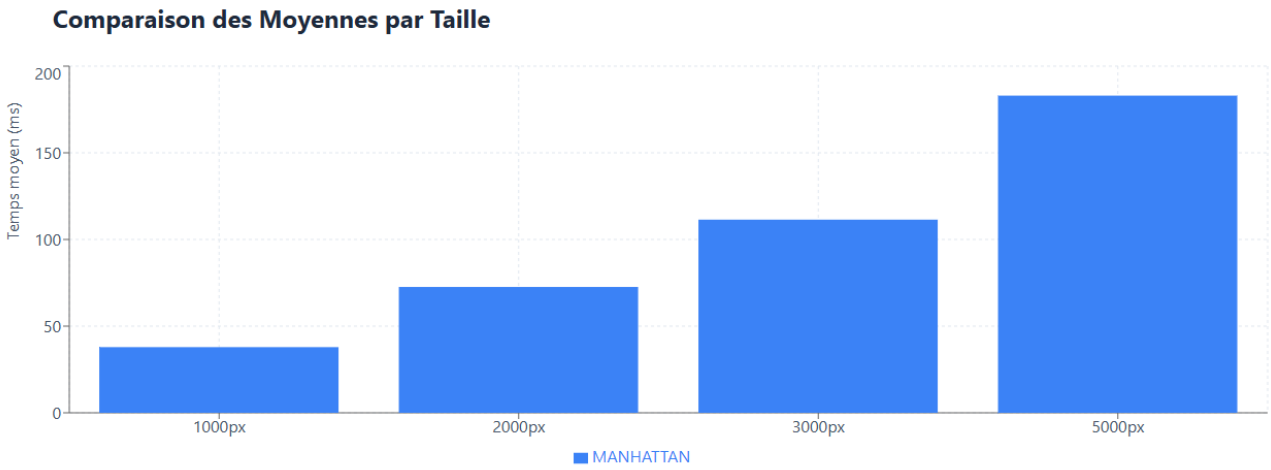


Figure 6: Comparaison de la méthode Manhattan avec BreakKey2 sur différente longueur d'image

La configuration BK2 pour le critère de Manhattan présente, à l'instar des autres tests sur cette plateforme, une efficacité radicalement supérieure avec une échelle de temps basculant en millisecondes (ms). Le traitement de 1000px s'effectue en moins de 40ms, tandis que la charge de 3000px est traitée en environ 115ms. Pour une image de 5000px, le temps reste stabilisé sous la barre des 190ms.

D) Corrélation croisée normalisé

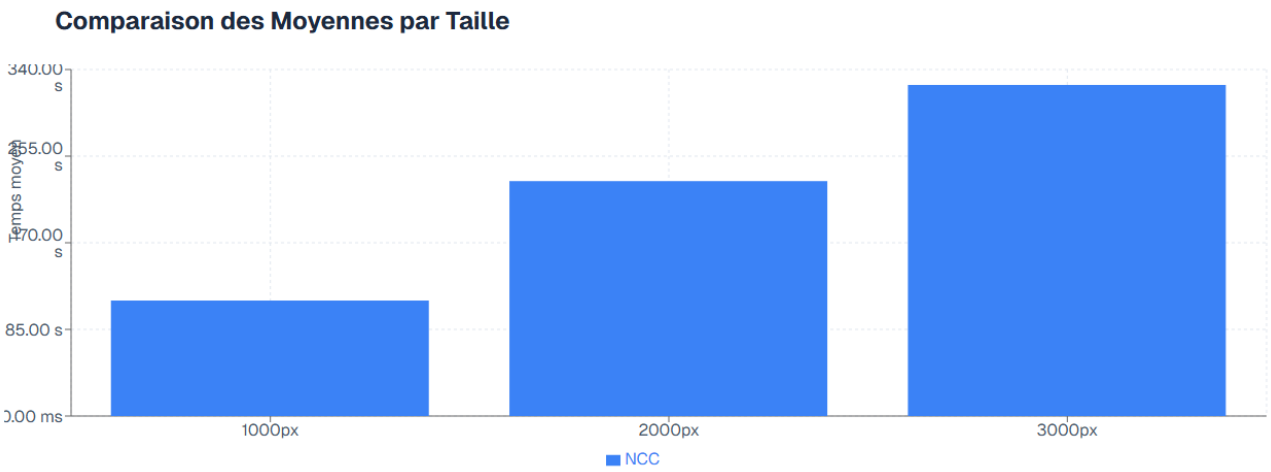


Figure 7: Comparaison de la méthode Corrélation croisée normalisé avec BreakKey1 sur différente longueur d'image

Le graphique de la configuration BK1 pour le critère NCC montre une progression linéaire du temps de calcul avec des valeurs extrêmement élevées. L'unité sur l'axe des ordonnées est la seconde (s). Pour une résolution de 1000px, le temps moyen est d'environ 115s, et il s'envole pour atteindre plus de 320s (environ 5 minutes et 20 secondes) pour 3000px. La corrélation croisée normalisée est mathématiquement lourde car elle nécessite, pour chaque décalage, des calculs de produits scalaires et de racines carrées pour la normalisation.

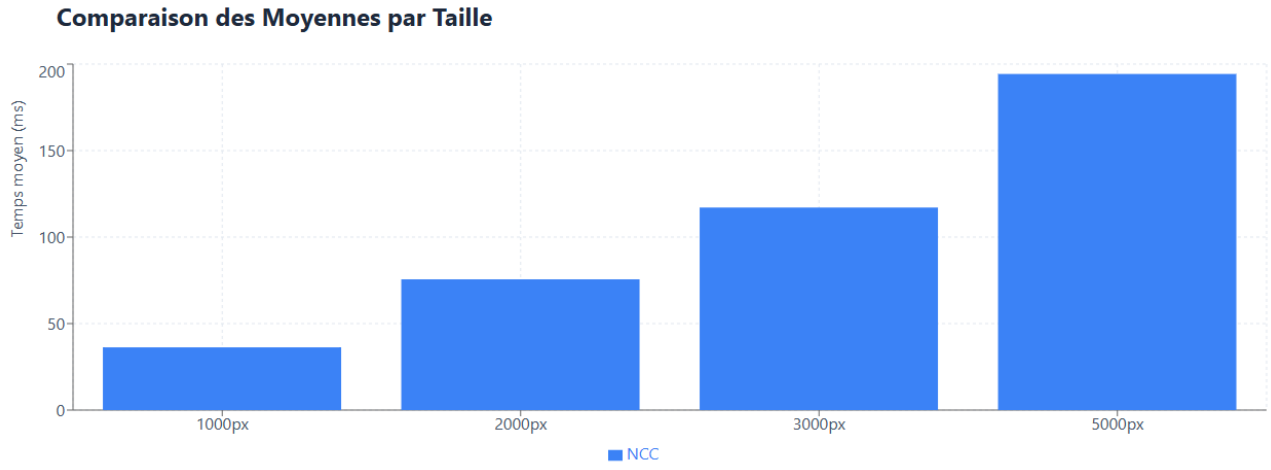


Figure 8: Comparaison de la méthode Corrélation croisée normalisé avec BreakKey2 sur différente longueur d'image

À l'opposé, la configuration BK2 pour le critère NCC affiche une efficacité de traitement remarquable avec une échelle en millisecondes (ms). Le temps moyen pour 1000px chute à environ 38ms, et il se stabilise autour de 118ms pour 3000px. Même en poussant le test à 5000px, le temps d'exécution reste contenu sous les 200ms.

E) Kulback-Leibler

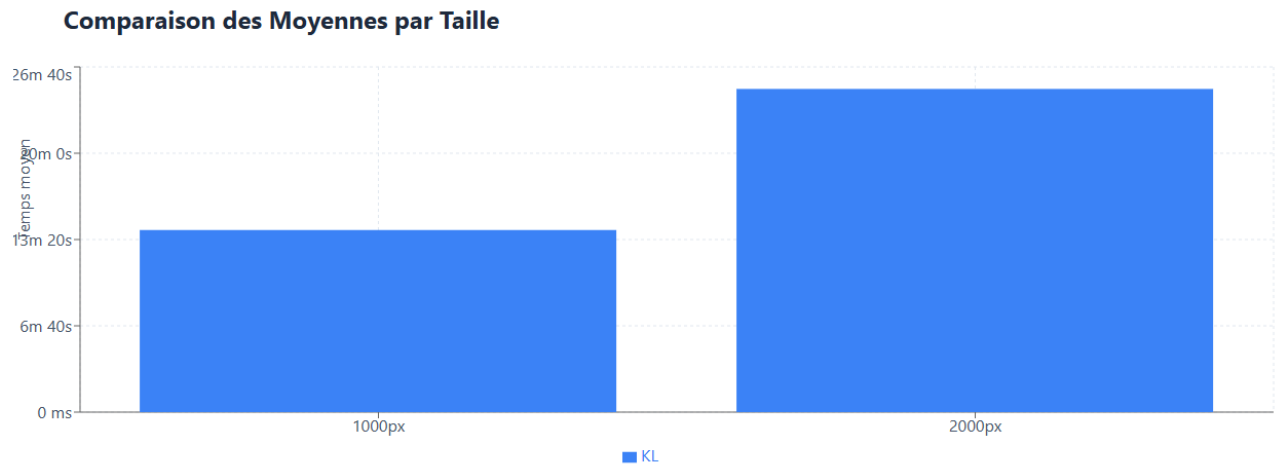


Figure 9: Comparaison de la méthode de Kulback-Leibler avec BreakKey1 sur différente longueur d'image

L'analyse du graphique BK1 pour le critère de Kullback-Leibler révèle des temps d'exécution critiques qui ne s'expriment plus en secondes, mais en minutes (m). Pour une résolution de 1000px, le temps moyen dépasse déjà les 13 minutes, et il double pour atteindre environ 26 minutes pour seulement 2000px. La divergence de KL implique des calculs de logarithmes et de rapports de probabilités pour chaque point de donnée, ce qui s'avère extrêmement lourd. Cette progression fulgurante démontre que sans optimisation majeure, ce critère est totalement impraticable pour un usage industriel ou temps réel dans l'environnement BK1.

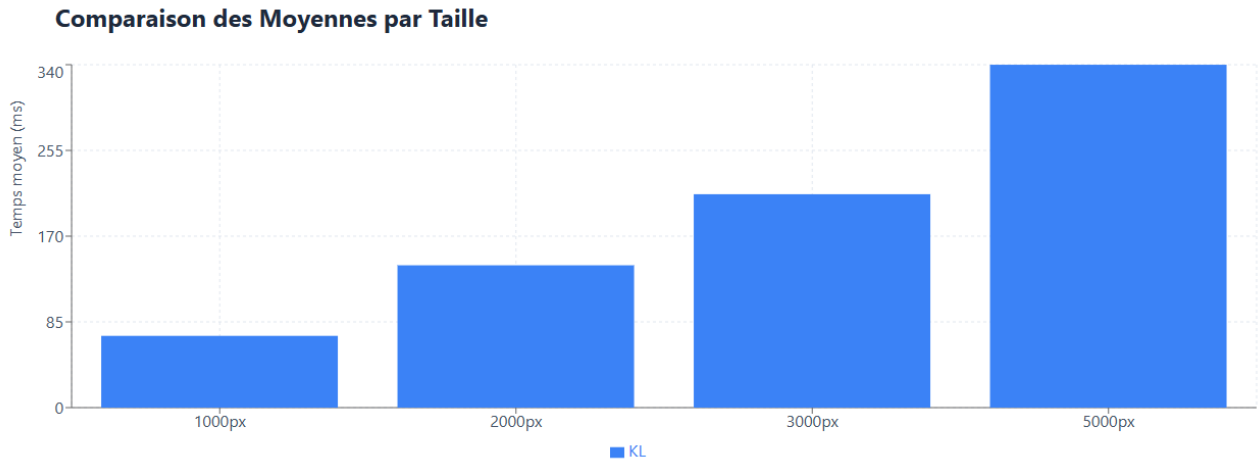


Figure 10: Comparaison de la méthode de Kulback-Leibler avec BreakKey2 sur différente longueur d'image

Le passage à la configuration BK2 montre, une fois de plus, une optimisation, ramenant les mesures à l'échelle des millisecondes (ms). Le traitement de 1000px est effectué en environ 70ms, celui de 3000px en environ 210ms, et même la charge maximale de 5000px est contenue sous les 350ms. Bien que KL reste le critère le plus lent sur la plateforme BK2 (comparé aux ~120ms d'Euclide pour 3000px), l'écart avec BK1 est vertigineux.

IV. Bilan

Cette étude comparative permet de conclure sur l'efficience des deux variantes algorithmiques développées. Le bilan repose sur deux axes majeurs, la performance brute (vitesse d'exécution) et la fiabilité opérationnelle (stabilité des temps).

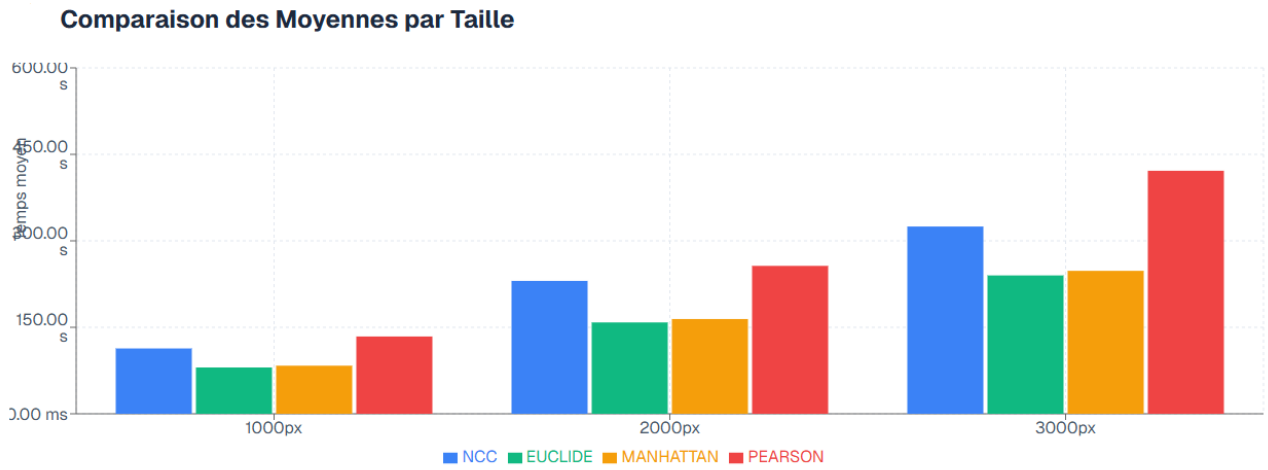


Figure 11: Comparaison des moyennes d'exécution pour BreakKey1 en fonction de la longueur de l'image

Cette variante affiche des temps de calcul s'élevant en secondes. Pour une image de 3000px, le critère de Pearson atteint par exemple un pic moyen de plus de 400s. Cette progression, bien que linéaire, démontre une saturation rapide des ressources, rendant BK1 inadapté aux environnements exigeant une réponse rapide.

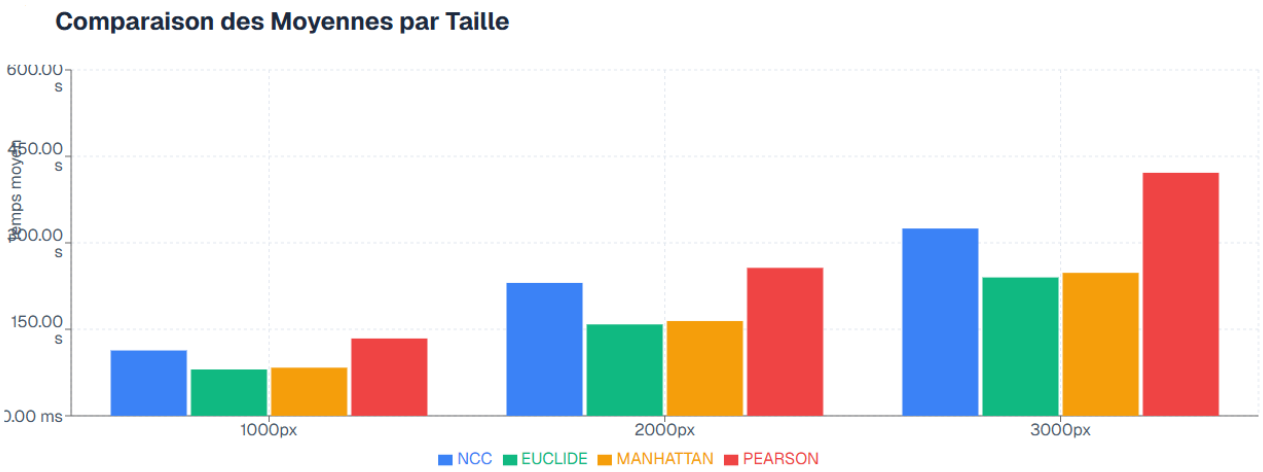


Figure 12: Comparaison des moyennes d'exécution pour BreakKey2 en fonction de la longueur de l'image

À l'inverse, BK2 opère une rupture d'échelle radicale en ramenant tous les calculs à l'ordre de la milliseconde (ms). Même pour des résolutions élevées de 5000px, les moyennes restent inférieures à 350ms pour l'ensemble des critères.

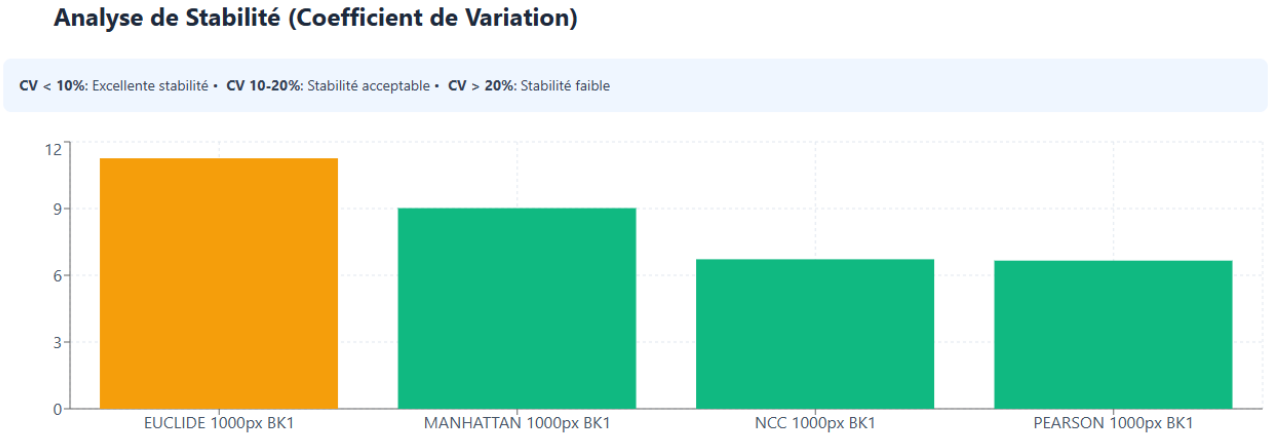


Figure 13: Graphique représentant l'analyse de la stabilité des différents critère avec BreakKey1 pour une longueur de 1000px

BK1 se distingue par une excellente stabilité. Les critères Manhattan, NCC et Pearson affichent des CV inférieurs à 10% (zone verte), garantissant un temps de réponse constant.

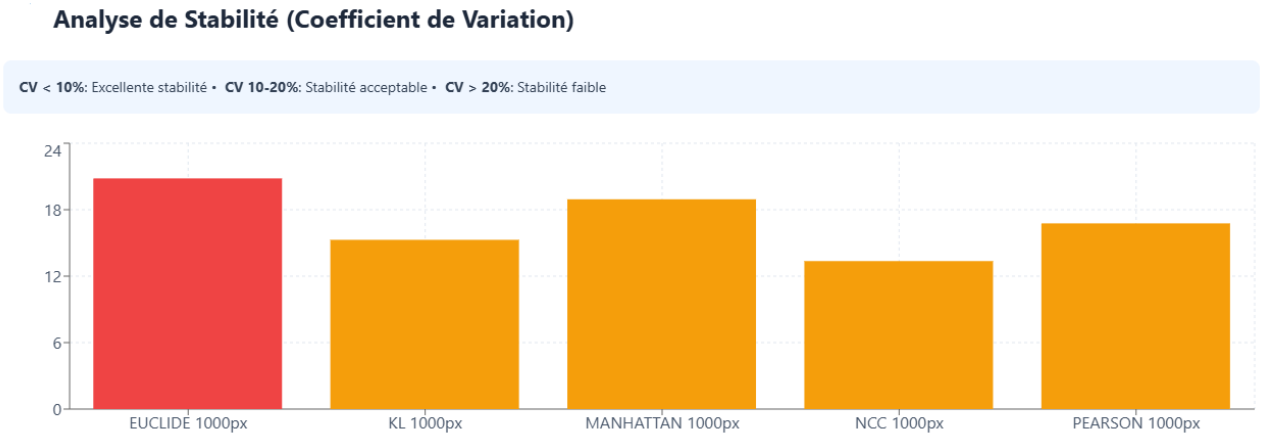


Figure 14: Graphique représentant l'analyse de la stabilité des différents critère avec BreakKey2 pour une longueur de 1000px

Malgré sa vitesse, BK2 présente une instabilité marquée, particulièrement sur le critère d'Euclide qui entre en "stabilité faible" avec un CV supérieur à 20% (zone rouge). Les autres critères oscillent dans une zone de stabilité simplement "acceptable". Ce phénomène est mathématiquement traduit par un écart-type de 185,22 ms, soit une dispersion supérieure à la moyenne elle-même (124,39 ms), ce qui prouve que la performance de BreakKey2 est extrêmement instable malgré sa rapidité.

L'étude comparative révèle une opposition radicale de comportement entre les deux approches analysées. D'un côté, BreakKey1 se caractérise par une exécution très lente, avec des temps moyens s'exprimant en secondes, mais offre une stabilité exemplaire avec un écart-type réduit. À l'opposé, BreakKey2 opère une rupture d'échelle en basculant dans l'ordre de la milliseconde , au prix d'une instabilité marquée.