

Image Denoising - TP3

Thomas Chabal

30 Octobre 2020

Table des matières

1	Non-Local Means Denoising	1
1.1	Résultats visuels	1
1.2	Analyse	1
2	Fast Approximation of Bilateral Filter using Discret Fourier Transform	6
2.1	Résultats visuels	6
2.2	Analyse	6
3	BM3D algorithm	12
3.1	Résultats visuels	12
3.2	Analyse	12

On évalue 3 algorithmes de suppression de bruit sur 3 images différentes : un fond noir uniforme, donc identique en tout point de l'image, un plateau de jeu d'échecs avec des bords saillants et également des nuances de marron pour le bois du plateau, ainsi qu'une photo de paysage où les textures des feuilles des arbres sont complexes (Figure 1).

1 Non-Local Means Denoising

1.1 Résultats visuels

1.2 Analyse

Ce papier se sert de l'auto-similarité dans les images pour les débruyter : on suppose qu'une image est suffisamment régulière pour qu'un même patch se répète en plusieurs zones presque identiquement, zones proches les unes des autres ou non. On débruite alors en moyennant les patchs identiques pour réduire le bruit en un même pixel : si un patch est présent n^2 fois dans



FIGURE 1 – Images évaluées sur les 3 algorithmes étudiés.

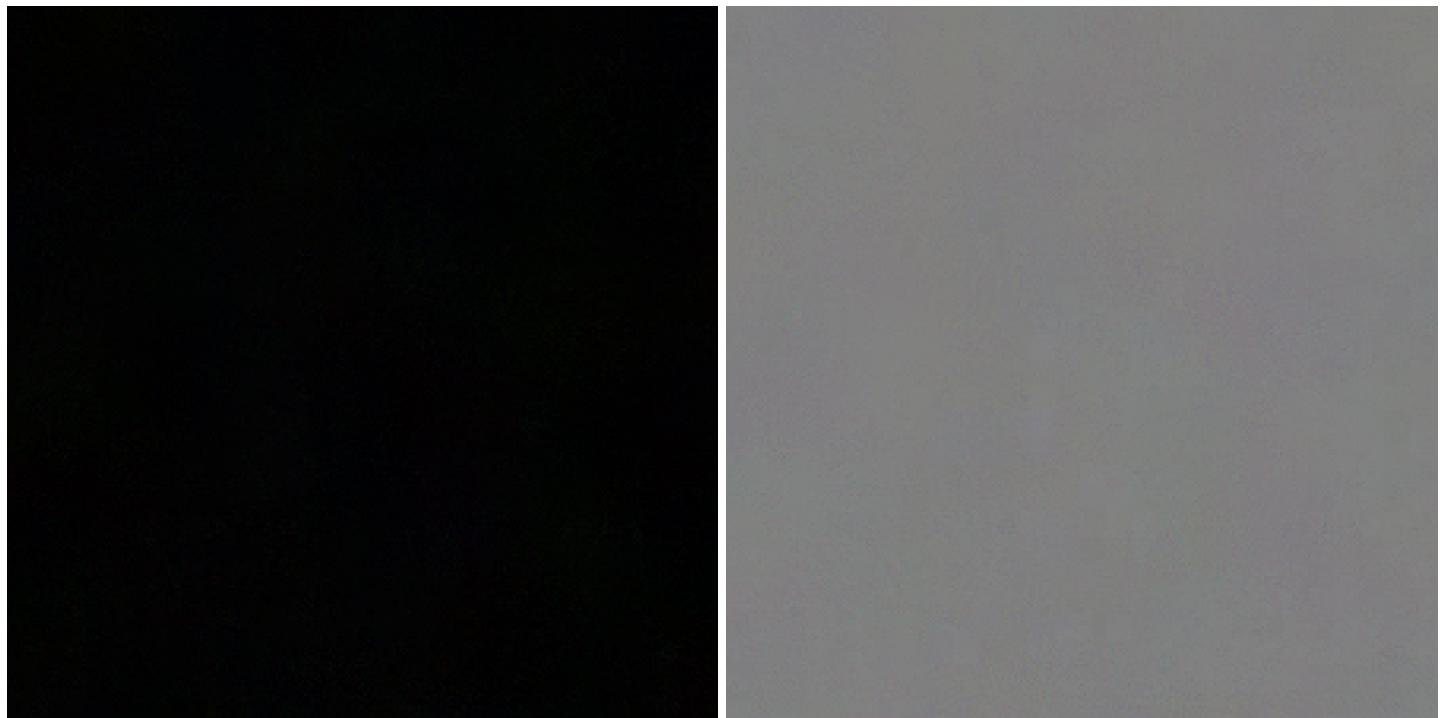


FIGURE 2 – Image noire uniforme débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 30$. Ce résultat est le même que pour $\sigma = 10$ ou $\sigma = 60$.

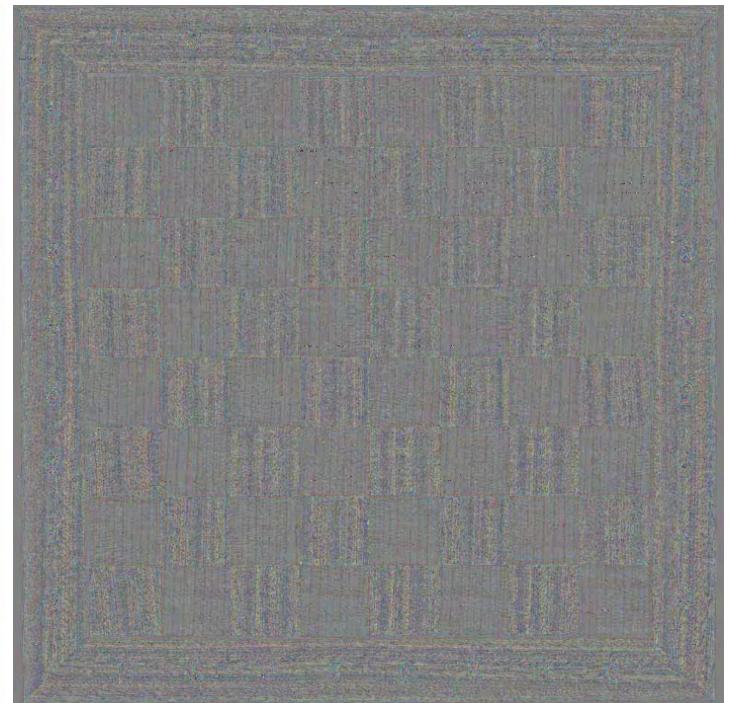


FIGURE 3 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 10$.

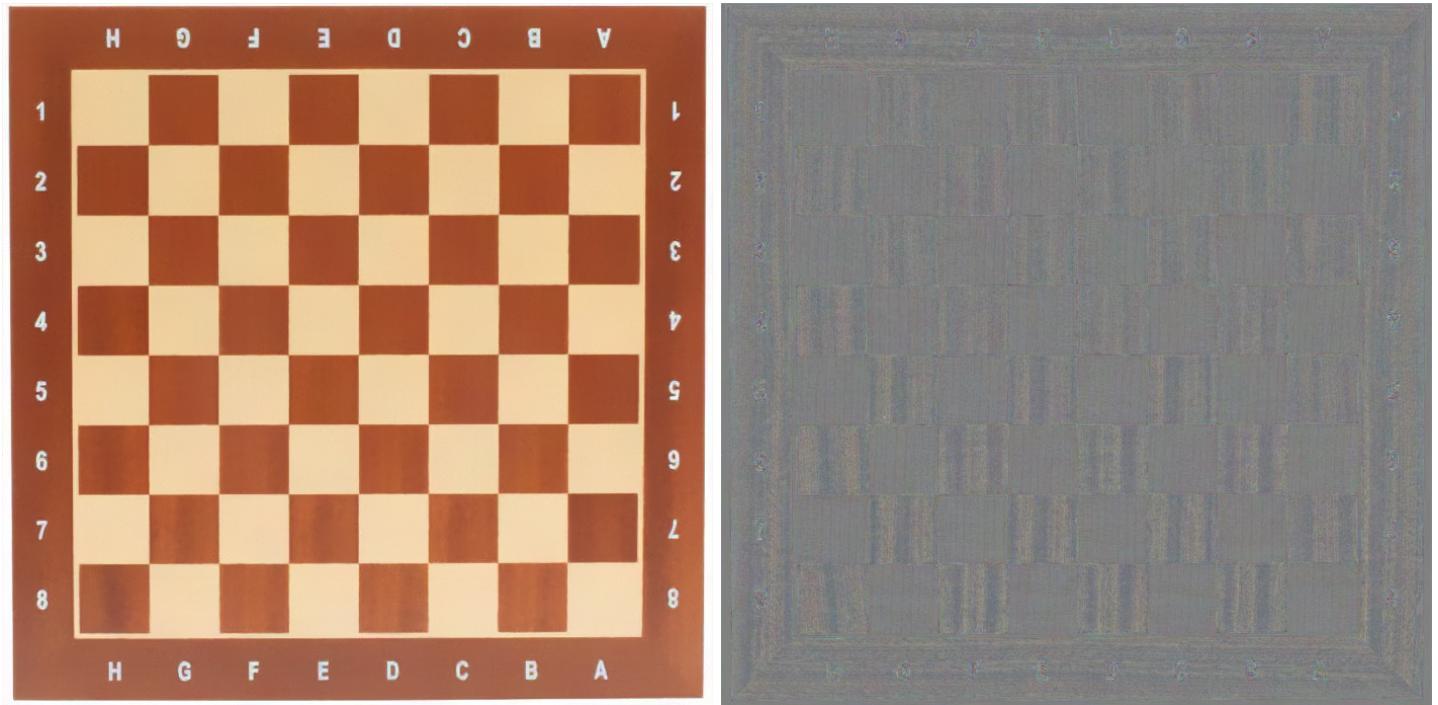


FIGURE 4 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 30$.

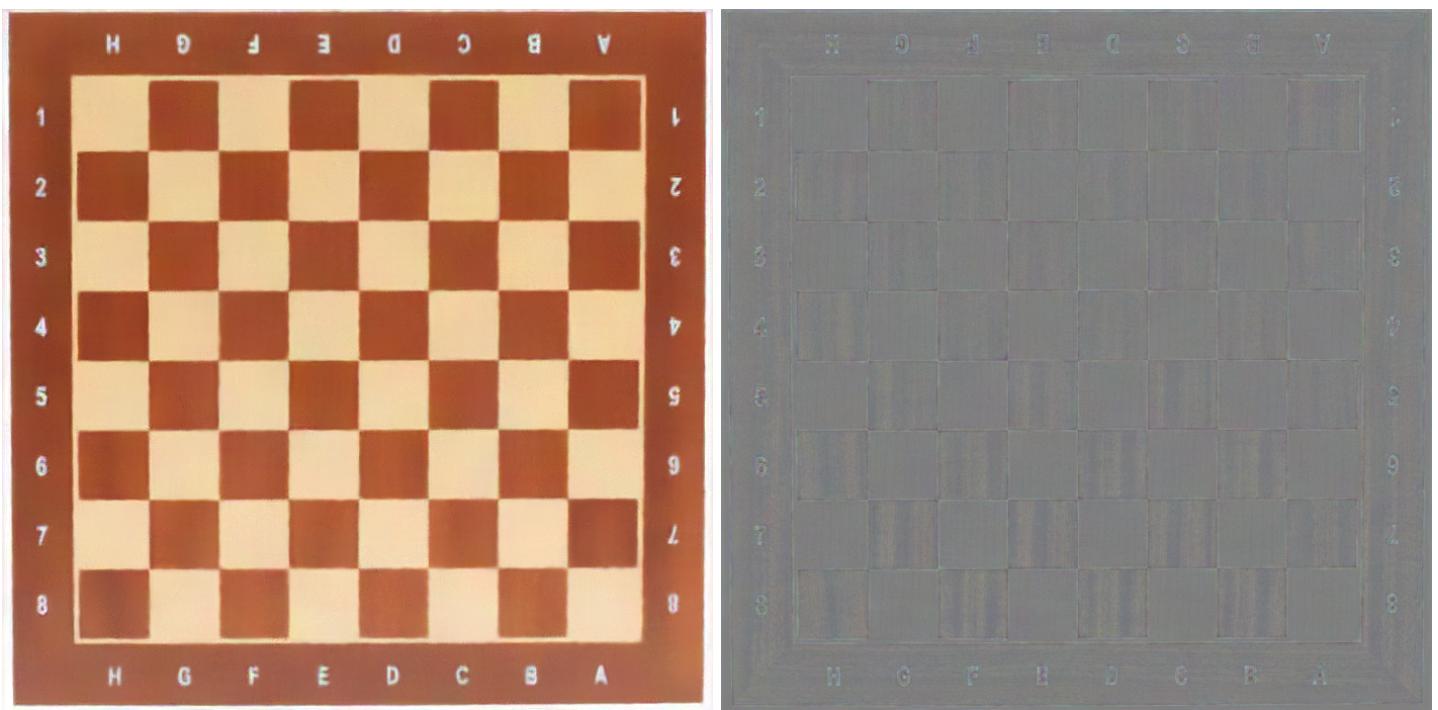


FIGURE 5 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 60$.



FIGURE 6 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour $\sigma = 10$.



FIGURE 7 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour $\sigma = 30$.

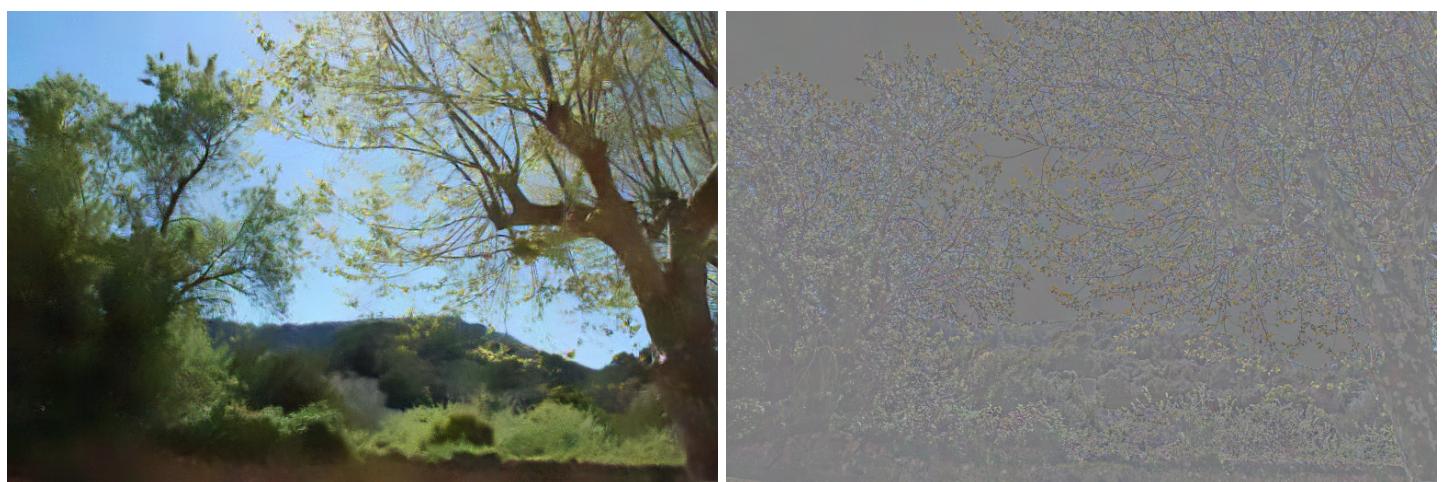


FIGURE 8 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour $\sigma = 60$.

Image	σ	RMSE	PSNR
Fond noir uniforme	10	1.03	47.88
	30	1.73	43.38
	60	2.45	40.34
Echecs	10	3.65	36.88
	30	5.47	33.37
	60	8.29	29.76
Arbre	10	8.50	29.54
	30	16.49	23.79
	60	24.28	20.43

TABLE 1 – RMSE et PSNR obtenu avec suppression de bruit pour chaque image selon la valeur σ du bruit initial.

l'image, alors le moyennage sur ces n^2 patchs réduira la variance de l'image par un facteur n et le bruit sera alors amoindri. Cela suppose cependant que l'image d'origine est auto-similaire, et des détails qui seraient présents sur certains des patchs identifiés risquent d'être supprimés lors du débruitage.

Ainsi, pour l'image uniforme noire, la théorie stipule que, l'image étant identique en tout point, le moyennage sera particulièrement efficace. On remarque effectivement que l'image débruitée est uniformément noire, et ce quelle que soit la variance σ du bruit blanc. La Table 1 montre d'ailleurs que pour les différentes variances de bruit, l'erreur carrée moyenne est de 1 ou 2 et le PSNR dépasse les 40dB, ce qui est synonyme d'excellent débruitage (les meilleurs algorithmes allant aux alentours de 35-37dB sur des images complexes).

Concernant le plateau d'échecs, pour des valeurs petites ou moyennes de σ , les bords des cases et les lettres sont parfaitement restitués. Cela n'est plus le cas pour une grande variance de bruit ($\sigma = 60$) où les bords des cases et des lettres deviennent plus flous et ne sont plus rectilignes. On observe de plus que le fond des cases, hétérogène en raison de l'aspect boiseux qui n'est pas uniforme, est retiré à mesure que σ augmente : on perd alors les détails du bois, correspondant à de très hautes fréquences. Quantitativement, le PSNR est très bon pour $\sigma = 10$ ($PSNR = 36.88$) mais il se détériore pour de plus grandes variances et n'est plus très bon pour $\sigma = 60$ ($PSNR \leq 30$).

Enfin, pour l'image naturelle, les textures sont mal débruitées par l'algorithme : visuellement, cela se remarque légèrement pour un léger bruit car les bords des feuilles ne sont plus très nets, et l'impact est de plus en plus visible à mesure que le bruit augmente. Là où la différence image originale - image bruitée fait apparaître un bruit plutôt uniforme pour $\sigma = 10$, cette différence fait ressortir la structure des arbres et de la végétation pour de plus fortes valeurs de σ . Quantitativement, l'erreur carrée est très élevée pour tout bruit ajouté et le PSNR devient mauvais ($PSNR \leq 25$ pour $\sigma \geq 30$).

On note aussi que plus la valeur du bruit ajouté est forte, plus le temps de calcul augmente (de façon linéaire avec σ).

En conclusion, cet algorithme fonctionne très bien pour des images régulières pour un bruit plus ou moins fort. Il conserve également la netteté des bords d'une image pour des bruits convenables, ce qui diffère de nombreuses méthodes faisant intervenir des noyaux gaussiens qui ont tendance à flouter les résultats finaux. A l'inverse, pour des images avec des détails nombreux, comme les feuilles d'un arbre, la méthode trouve ses limites car l'image n'est plus auto-semblable. On perd alors de nombreuses informations dans l'image : il faut donc envisager d'autres méthodes.

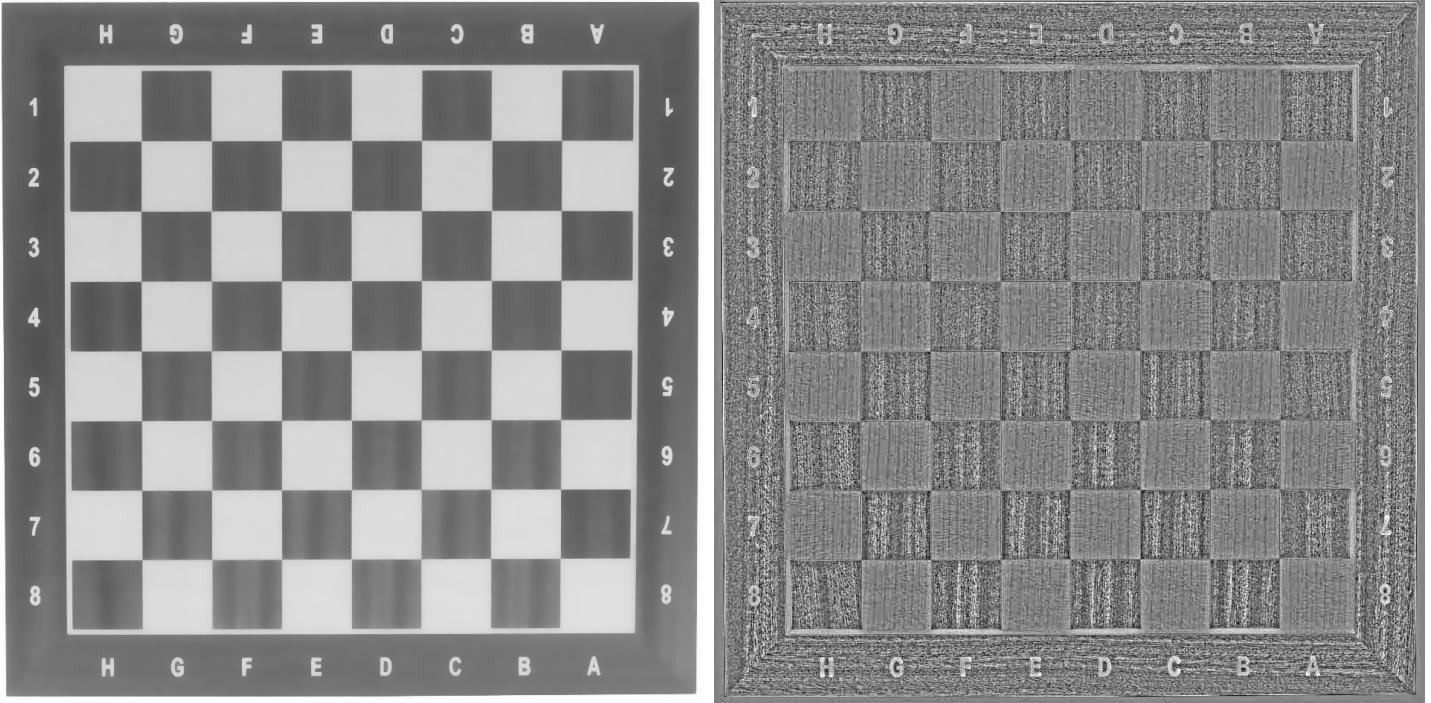


FIGURE 9 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.001$.

2 Fast Approximation of Bilateral Filter using Discret Fourier Transform

2.1 Résultats visuels

2.2 Analyse

Ce papier implémente une méthode rapide d'approximation de filtre bilatéral en ayant recours à des transformées de Fourier discrètes. Le calcul du filtre bilatéral nécessite d'effectuer de nombreuses convolutions de l'image avec des gaussiennes sur des patchs de tailles variables, ce qui est très coûteux en temps de calcul. A l'inverse, l'approximation des gaussiennes par une transformation discrète et tronquée de Fourier permet de simuler fidèlement (avec un seuil de tolérance ϵ) les gaussiennes et ainsi faciliter le calcul du filtre en permettant de paralléliser les opérations et de ne pas recalculer tous les filtres à chaque itération de l'algorithme. La coupure de la DFT est effectuée de sorte que la plupart des informations de la gaussienne sont incluses dans les coefficients de la DCT, mais les très hautes fréquences sont retirées ce qui risque d'impacter la suppression de bruit dans les images.

Pour l'image uniforme, le débruitage fonctionne toujours très bien. Le calcul se fait en moins de 100 millisecondes et fournit dans tous les cas une image débruitée uniforme : l'algorithme semble être efficace pour les zones plates.

Pour le plateau d'échecs, lorsque σ_S augmente, l'intérieur des cases est atténué mais les bords de l'image restent saillants. Lorsque σ_R croît, le résultat global paraît très flou et les bords ne sont plus aussi nets qu'avant, mais le motif intérieur des cases, plus régulier, ne semble pas trop impacté. Enfin, pour une tolérance ϵ plus forte, les bords de l'image sont plus flous.

Concernant le paysage, les résultats de débruitage sont mauvais pour tous les paramètres : les hautes fréquences ont été retirées et les détails de l'image ont donc été effacés de l'image, c'est-à-dire que les feuilles ne sont plus distinguables. On peut noter que σ_S semble impacter le contraste de l'image, qui est plus faible pour une valeur plus élevée de σ_S , tandis que σ_R et ϵ impactent directement le caractère flou de la photo. On observe d'ailleurs que l'image n'est pas identiquement floue : certaines zones sont particulièrement impactées alors que d'autres voisines restent très nettes. Sur toutes les différences *imageorigine*–*imagedebruitee*, la structure de l'image ressort clairement, ce qui signifie que les détails ne sont pas correctement éliminés.

Enfin, l'étude des coefficients DFT et des temps d'exécution fait ressortir que les paramètres influent différemment sur les temps de calculs. Pour une variation de σ_S , le temps reste constant et la décomposition DFT effectuée est la même. Lorsque σ_R double, l'algorithme requiert un calcul d'un plus grand nombre de coefficients DFT et le temps de calcul peut doubler pour des images non uniformes. Il est donc important de sélectionner une variance du filtre bilatéral qui soit assez faible. Enfin, l'augmentation de la tolérance fait décroître significativement le nombre de coefficients DFT nécessaires pour approximer la

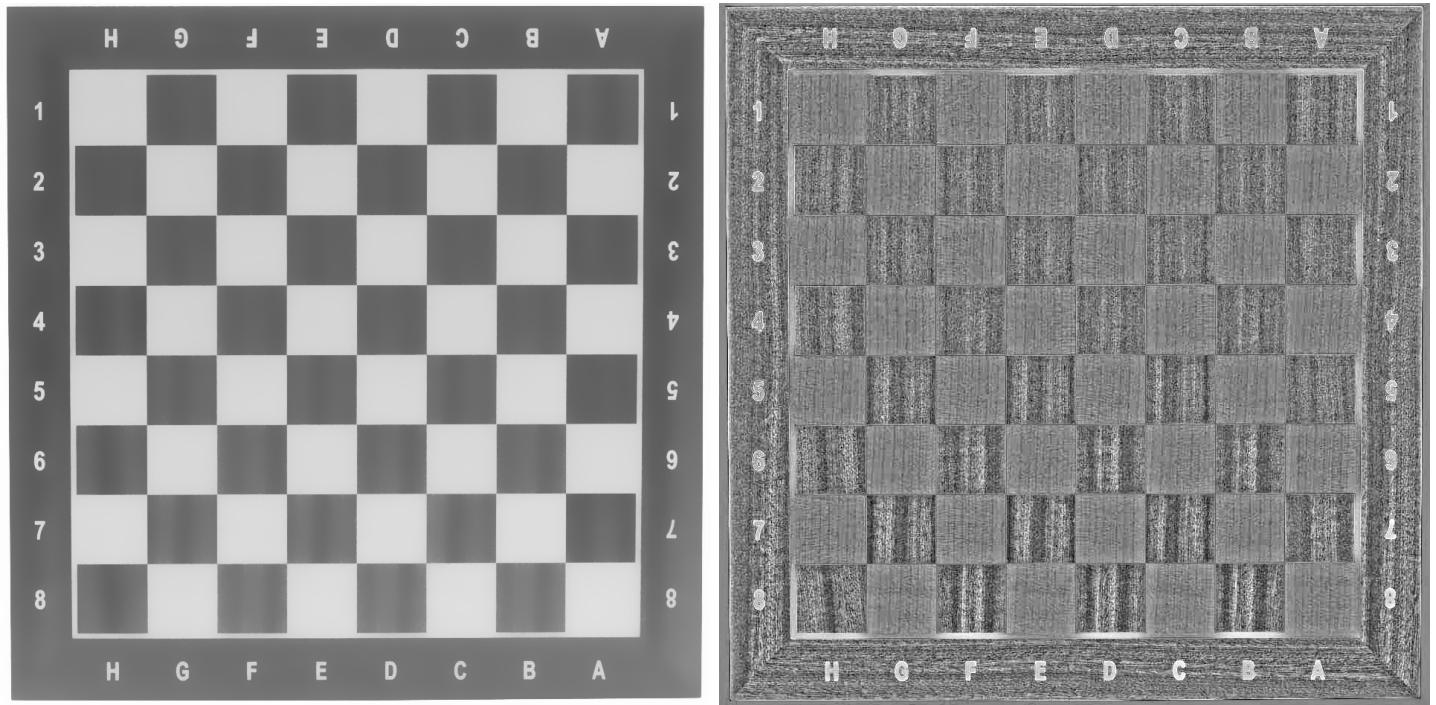


FIGURE 10 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 10$, $\sigma_R = 40$ et $\epsilon = 0.001$.

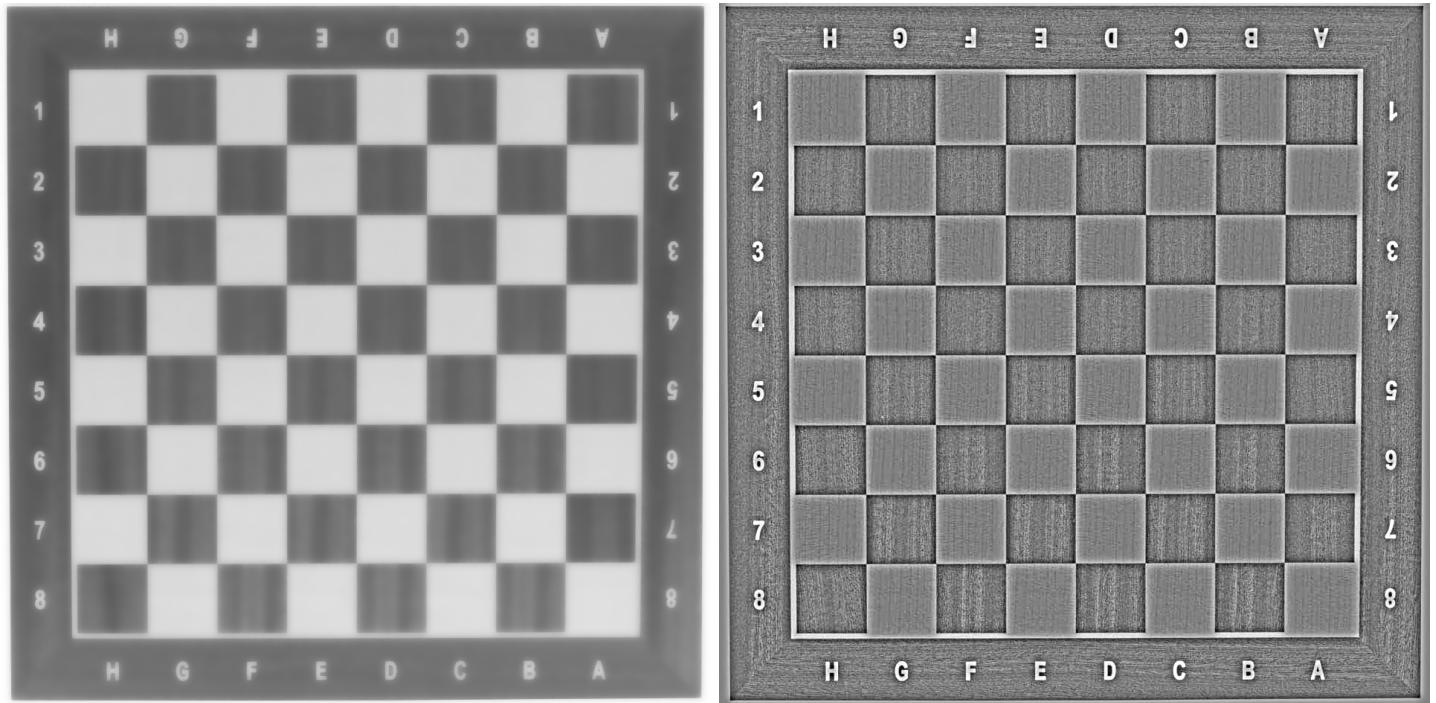


FIGURE 11 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 80$ et $\epsilon = 0.001$.

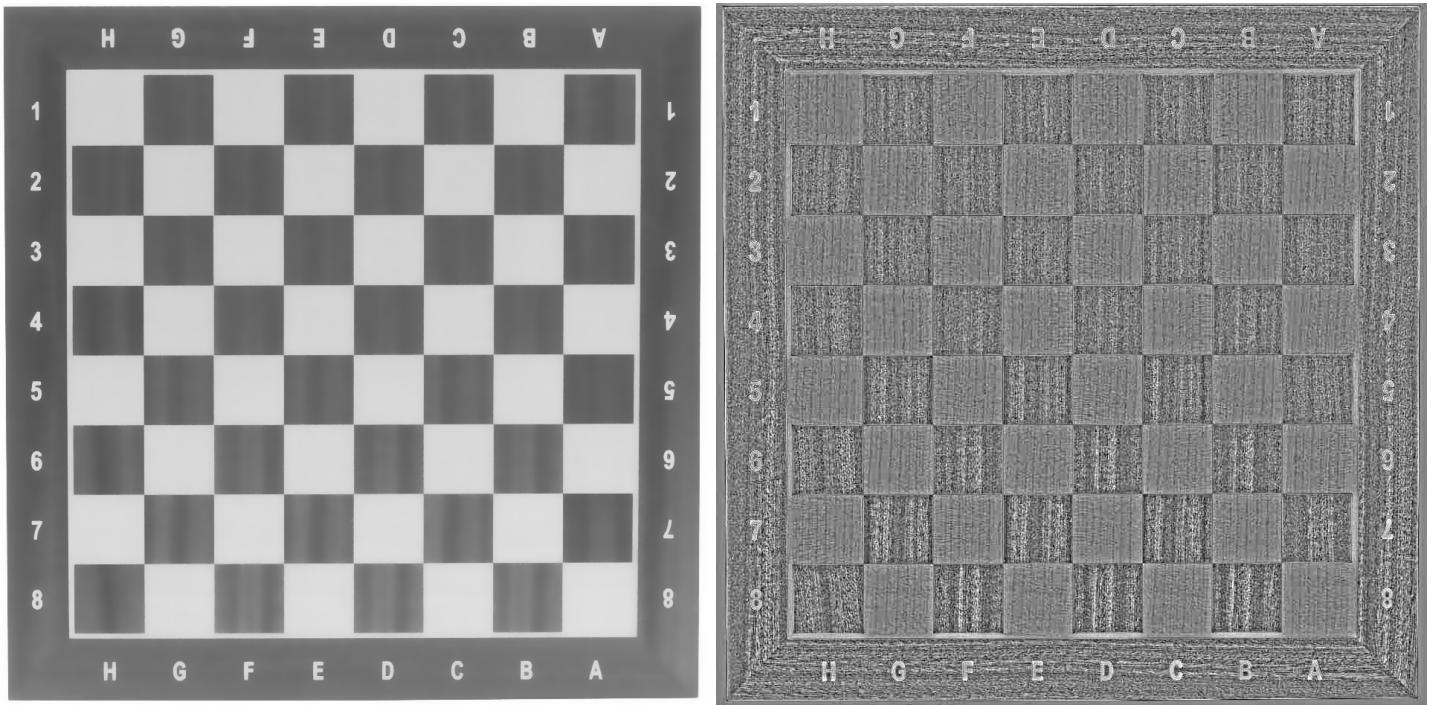


FIGURE 12 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.01$.

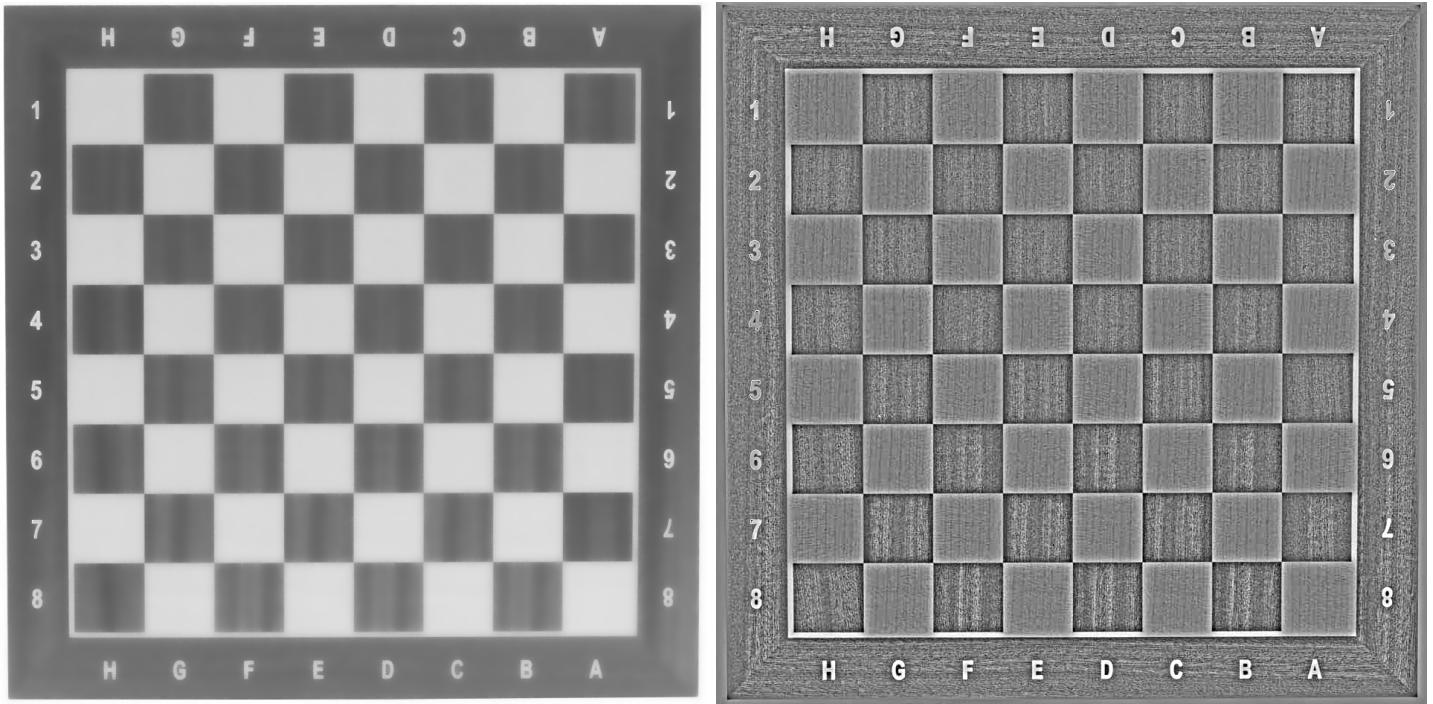


FIGURE 13 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.5$.

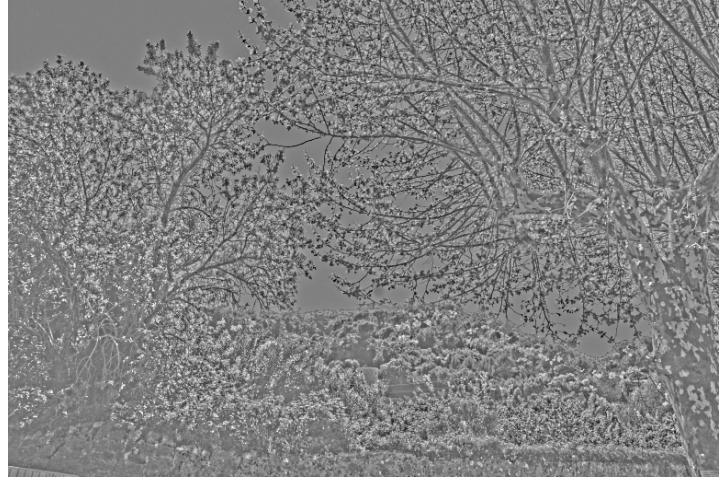


FIGURE 14 – Photo de végétation débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.001$.

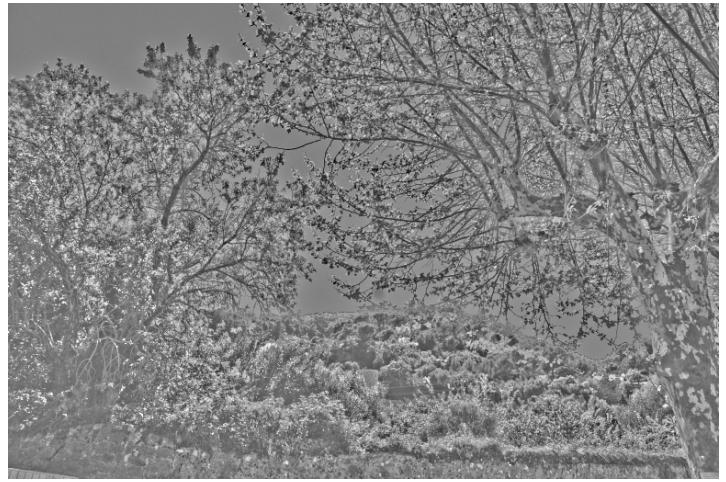


FIGURE 15 – Photo de végétation débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 10$, $\sigma_R = 40$ et $\epsilon = 0.001$.

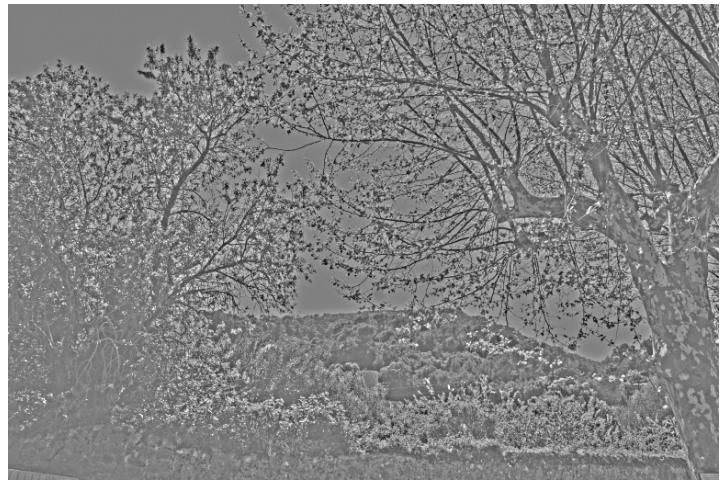


FIGURE 16 – Photo de végétation débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 80$ et $\epsilon = 0.001$.

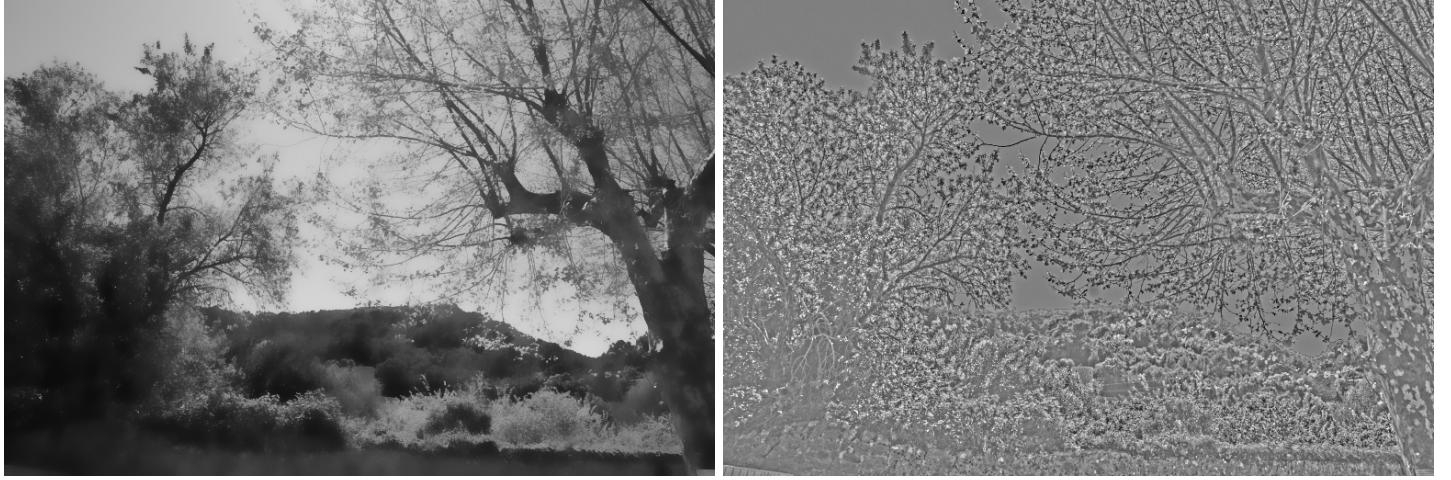


FIGURE 17 – Photo de végétation débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.01$.

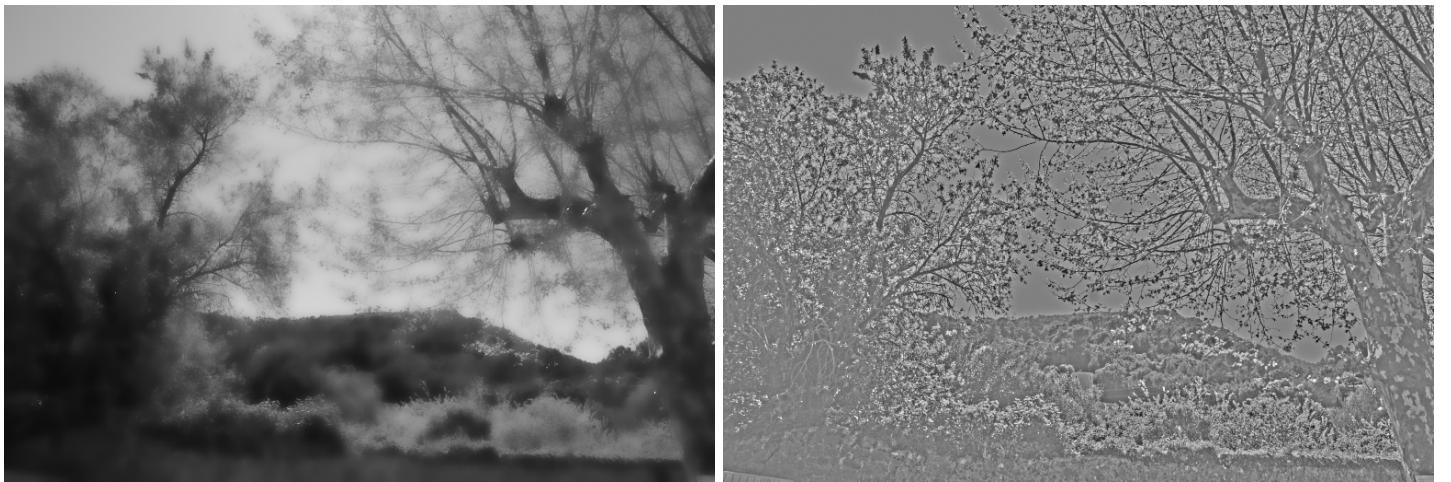


FIGURE 18 – Photo de végétation débruitée à gauche et sa différence avec l'image bruitée à droite, pour $\sigma_S = 5$, $\sigma_R = 40$ et $\epsilon = 0.5$.

Image	σ_S	σ_R	ϵ	Number of DFT coefficients	Time (ms)
Fond noir uniforme	5	40	0.001	11	88
	10	40	0.001	11	96
	5	80	0.001	12	93
	5	40	0.01	4	76
	5	40	0.5	2	70
Echecs	5	40	0.001	6	481
	10	40	0.001	6	550
	5	80	0.001	12	1028
	5	40	0.01	5	534
Arbre	5	40	0.001	7	260
	10	40	0.001	7	263
	5	80	0.001	12	473
	5	40	0.01	6	260
	5	40	0.5	2	254

TABLE 2 – Nombre de coefficients DFT nécessaires pour approximer les gaussiennes et temps d'exécution de l'algorithme fast bilateral en fonction des σ_S , σ_R et ϵ .

gaussienne, ce qui est cohérent avec le fait que les coefficients plus élevés dans la DFT correspondent à des hautes fréquences qui ne sont pas toujours nécessaire pour avoir la forme globale du signal. Le temps de calcul reste inchangé.

En conclusion, cet algorithme fait gagner beaucoup de temps pour effectuer un calcul d'approximation de filtre bilatéral par rapport à un filtre bilatéral classique. En revanche, les performances sont visuellement plutôt mauvaises pour les détails et textures d'une image qui ne sont pas très bien restituées après débruitage.

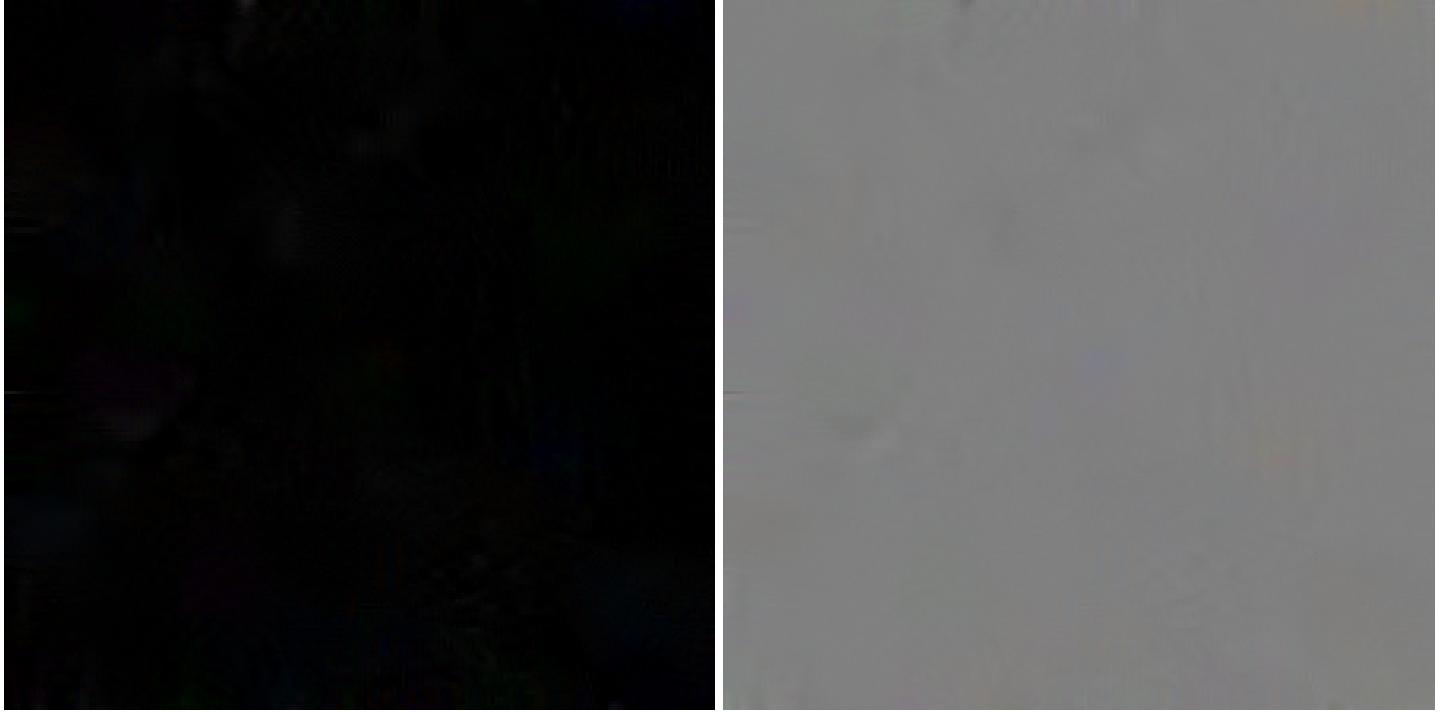


FIGURE 19 – Fond uniforme débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 60$.

Image	σ	RMSE	PSNR	Time (s)
Fond noir uniforme	10	0.62	52.3	4.47
	30	1.71	43.5	4.77
	60	2.67	39.6	5.48
Echecs	10	3.25	37.9	16.41
	30	5.25	33.7	20.71
	60	7.22	31	25.37
Arbre	10	7	31.2	8.69
	30	15	24.6	10.25
	60	24.2	20.4	13.22

TABLE 3 – Temps d'exécution, RMSE et PSNR pour le débruitage des images en fonction de σ .

3 BM3D algorithm

3.1 Résultats visuels

3.2 Analyse

On observe que les résultats de l'algorithme BM3D sont globalement très encourageants.

Dans le cas d'un fond noir uniforme, le résultat est très bon pour un bruit de variance 10 ou 30 et il faut attendre $\sigma = 60$ pour voir des artefacts apparaître (des zones plus grises foncées, surtout visibles sur la différence d'images). Le temps de calcul est cependant de 5 secondes, ce qui est très lent (d'autant plus que l'image est de petite dimension).

Pour le plateau d'échecs, le débruitage est excellent pour $\sigma = 10$: le contenu de l'image ne semble pas impacté, la texture du bois est toujours présente et la différence d'images fait surtout ressortir un bruit uniforme gaussien. Pour un bruit plus important ($\sigma = 30$), le résultat est toujours très satisfaisant, même si certaines portions boiseuses ont perdu en détails de texture (il suffit par exemple de regarder le coin supérieur gauche du plateau qui comporte moins de stries). Pour un bruit élevé ($\sigma = 60$), l'image est toujours très nette, plus jolie qu'avec les algorithmes précédents, mais des artefacts sont tout de même très présents sur les zones uniformes et celles de bois du plateau. Le temps de calcul est également très long : entre 15 et 25 secondes pour débruiter une seule image.

Enfin, concernant le paysage, le résultat est surprenant pour un bruit faible : les détails des feuilles sont extrêmement

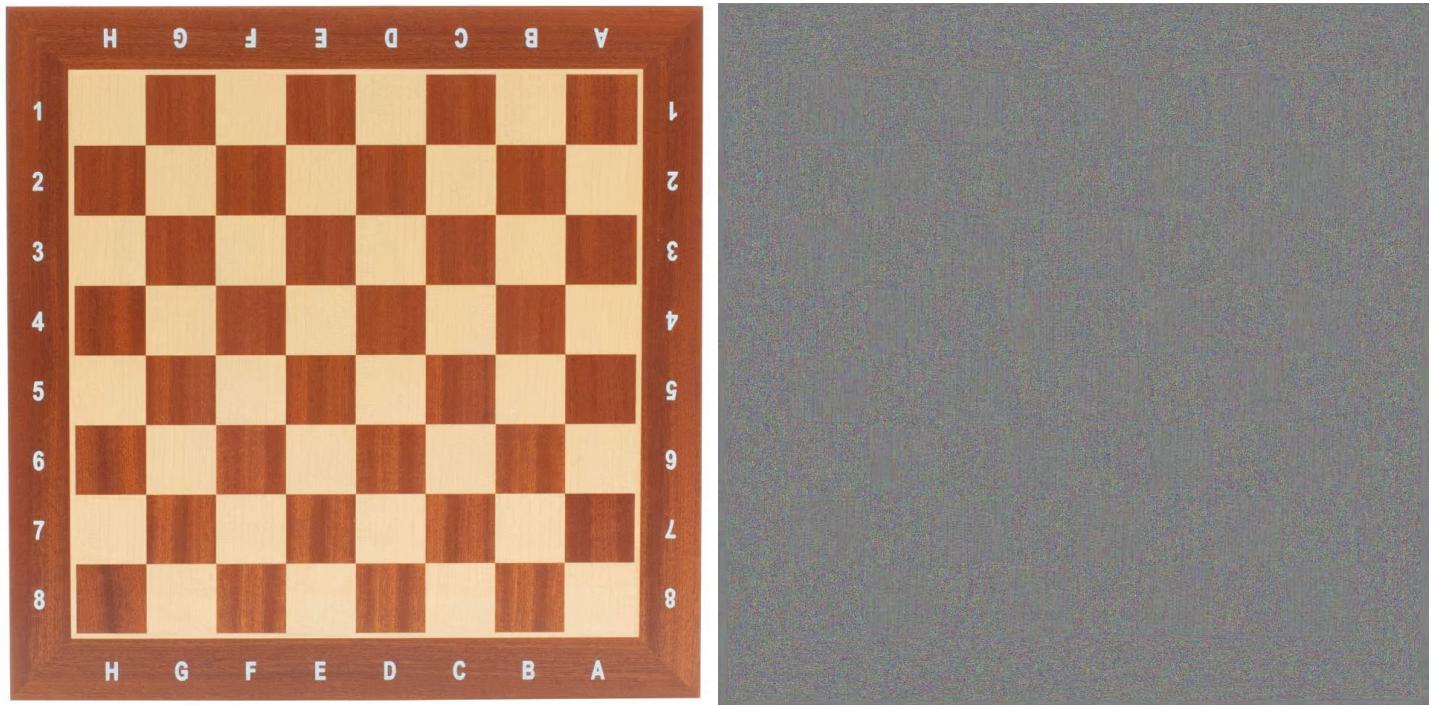


FIGURE 20 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 10$.

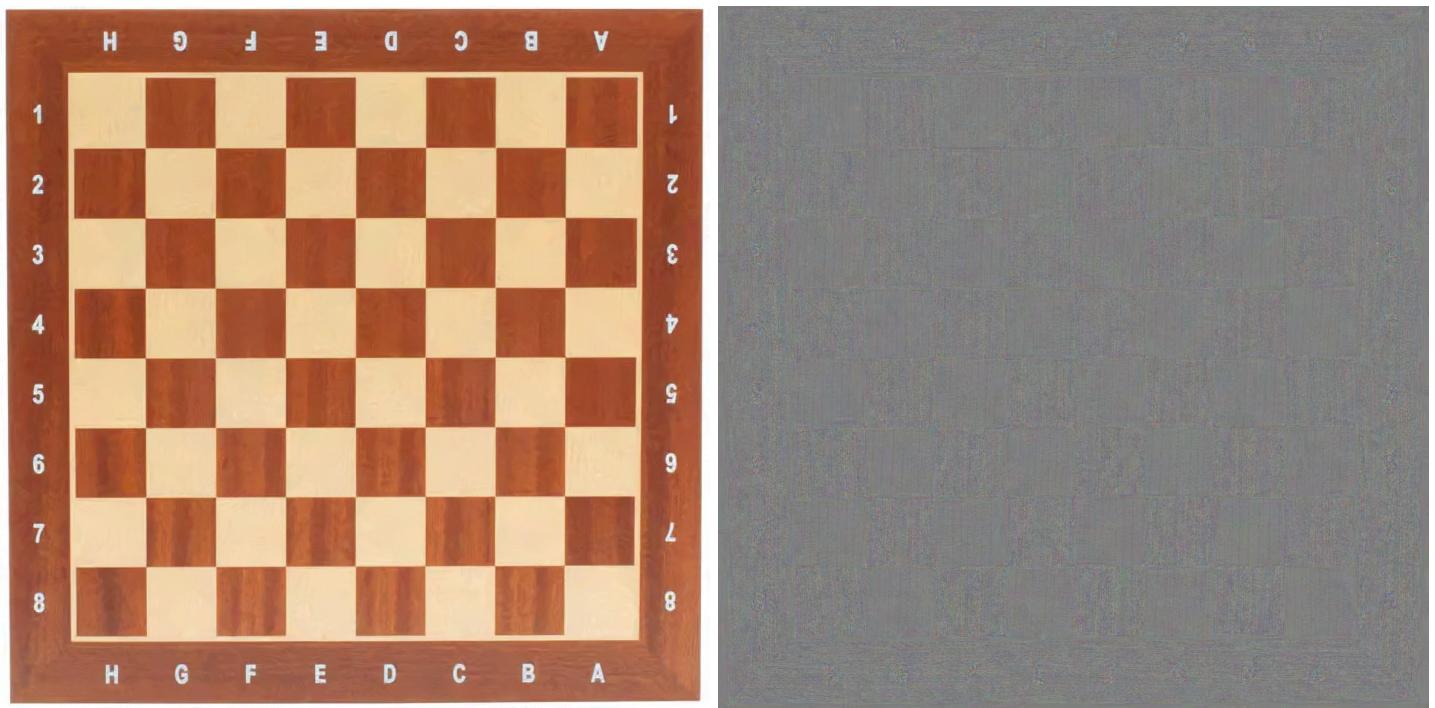


FIGURE 21 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 30$.

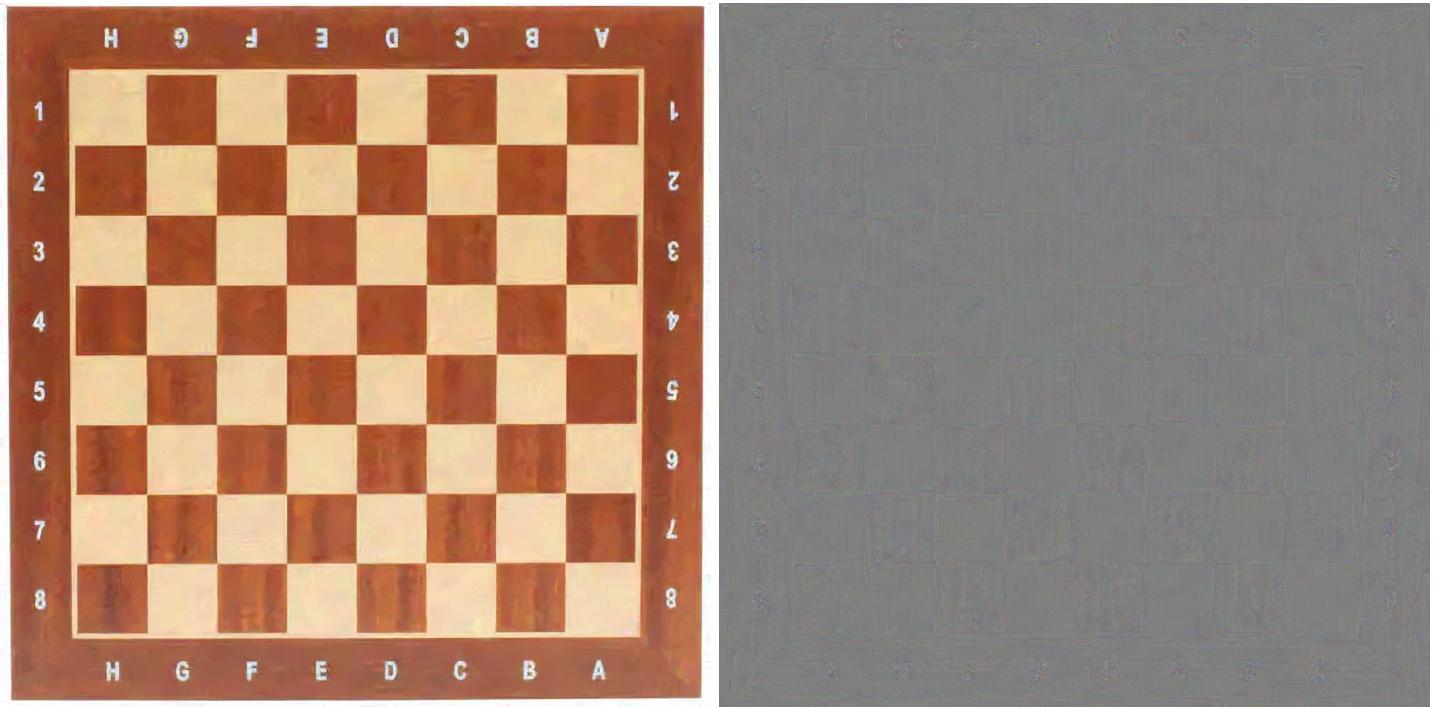


FIGURE 22 – Plateau d'échecs débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 60$.



FIGURE 23 – Paysage débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 10$.



FIGURE 24 – Paysage débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 30$.

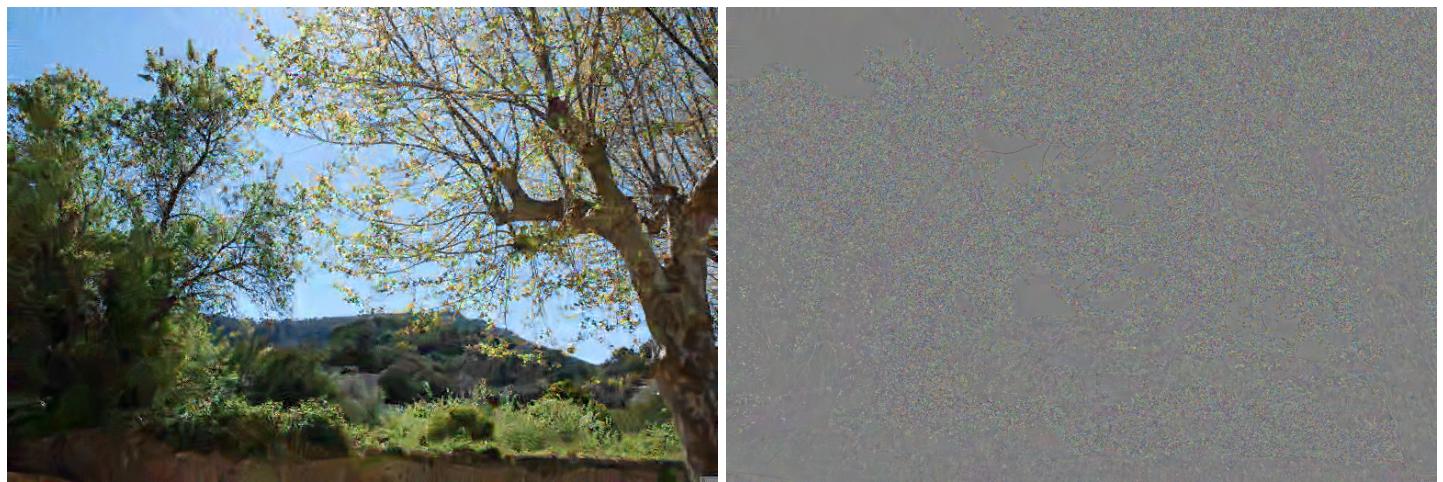


FIGURE 25 – Paysage débruité à gauche et sa différence avec l'image bruitée à droite, pour $\sigma = 60$.

bien restitués et l'image est très nette. La différence avec l'image originale montre d'ailleurs un bruit gaussien uniforme : le débruitage a été très efficace. Pour un bruit intermédiaire ($\sigma = 30$), l'image débruitée est toujours satisfaisante (les bords semblent toujours plutôt nets) même si certaines zones ont perdu de l'information, comme par exemple les arbres plus sombres au centre de la photo dont on ne voit plus les feuilles et branches. La différence d'images pour cet exemple montre également un bruit important. En revanche, pour un bruit fort ($\sigma = 60$), le résultat est très détérioré et on observe l'apparition de nombreuses aberrations chromatiques autour des feuilles, ainsi que des artefacts comme au bord supérieur gauche de l'image. Cette fois-ci, la différence avec l'image originale fait ressortir quelques branches et structures des arbres, ce qui explique que le débruitage est mauvais.

Les résultats restent meilleurs qu'avec les autres algorithmes pour un bruit faible ou modéré, et se détériorent avec un bruit plus fort. Les temps de calcul sont considérablement accrus : on met désormais entre 5 secondes pour une image uniforme et 25 secondes pour un plateau d'échecs dans le calcul de l'image nettoyée. Le PSNR est tout de même très satisfaisant puisqu'on dépasse dans la plupart des cas les 30dB et que l'on s'approche des 35dB pour un bruit gaussien de variance inférieure à 30. Il s'agit donc bel et bien de l'algorithme le plus performant dans la suppression de bruit pour une image.