

# Image Denoising - TP4

Thomas Chabal

6 Novembre 2020

## 1 Implementation of the Non-Local Bayes (NL-Bayes) Image Denoising Algorithm

Les 3 images sur lesquelles sont évaluées l'article *Implementation of the Non-Local Bayes (NL-Bayes) Image Denoising Algorithm* sont les suivantes : une photo de fond uniforme noir, une photo de plateau d'échecs et une photo de paysage.

D'abord, pour une image uniformément noire, l'algorithme n'a quasiment jamais de problème pour débruiter l'image et rendre un résultat identique à l'image d'origine, et ce dès la première étape de l'algorithme. Le PSNR est d'ailleurs quasiment toujours supérieur à 43, ce qui est synonyme d'excellence. Un mauvais débruitage intervient seulement dans le cas où aucun critère de zones homogènes n'est pris en compte (ni à la première étape, ni à la deuxième) : on voit alors apparaître des zones grises très foncées (difficiles à distinguer mais tout de même présentes un peu partout dans l'image) et le PSNR chute à 34.

Ensuite, sur la photo d'échiquier, le bois des cases marron est un élément difficile à conserver lors du débruitage. Lorsque le bruit est de variance faible ( $\sigma = 10$ ), le résultat est excellent, on conserve tous les détails du damier et les bords restent parfaitement nets. Lorsque le bruit est de variance  $\sigma = 30$ , la première étape de débruitage a tendance à retirer beaucoup de bruit et lisser l'image tandis que la seconde étape rajoute des artefacts et détériore le résultat final. Puisque l'image n'est pas assez auto-similaire, l'algorithme qui est non localisé a tendance à flouter certaines zones où la texture du bois varie. Si on effectue un calcul en prenant en compte le critère d'homogénéisation des zones lors des deux étapes, le résultat obtenu est presque identique à celui évoqué juste au-dessus, par contre celui-ci change si on ne prend en compte ce critère dans aucune étape de débruitage : les zones boiseuses voient alors apparaître de l'hétérogénéité et la texture est légèrement plus proche de celle du bois, mais toujours pas idéale. Enfin, lorsque l'on calcule le biais de l'image, l'algorithme fait surtout attention aux bords dans l'image comme on le voit dans la figure 8, mais visuellement le résultat ne change pas. Dans toutes ces situations, les zones uniformes et beige du damier ont vu apparaître du bruit : on a perdu en qualité d'image, et le PSNR le montre puisqu'il est tombé à 33 (au lieu de 37 pour  $\sigma = 10$ ). Lorsque le bruit est fort ( $\sigma = 60$ ), il est a priori très compliqué voire impossible de restituer les textures d'origine du bois puisque qu'elles sont effacées par le bruit. De fait, la première étape de l'algorithme uniformise l'image pour en retirer le bruit puis la seconde étape détériore la qualité en rajoutant des détails de façon inhomogène dans l'image, ce qui casse la cohérence de l'image (certaines cases sont uniformes tandis que d'autres voient



FIGURE 1 – Images évaluées sur les 3 algorithmes étudiés.

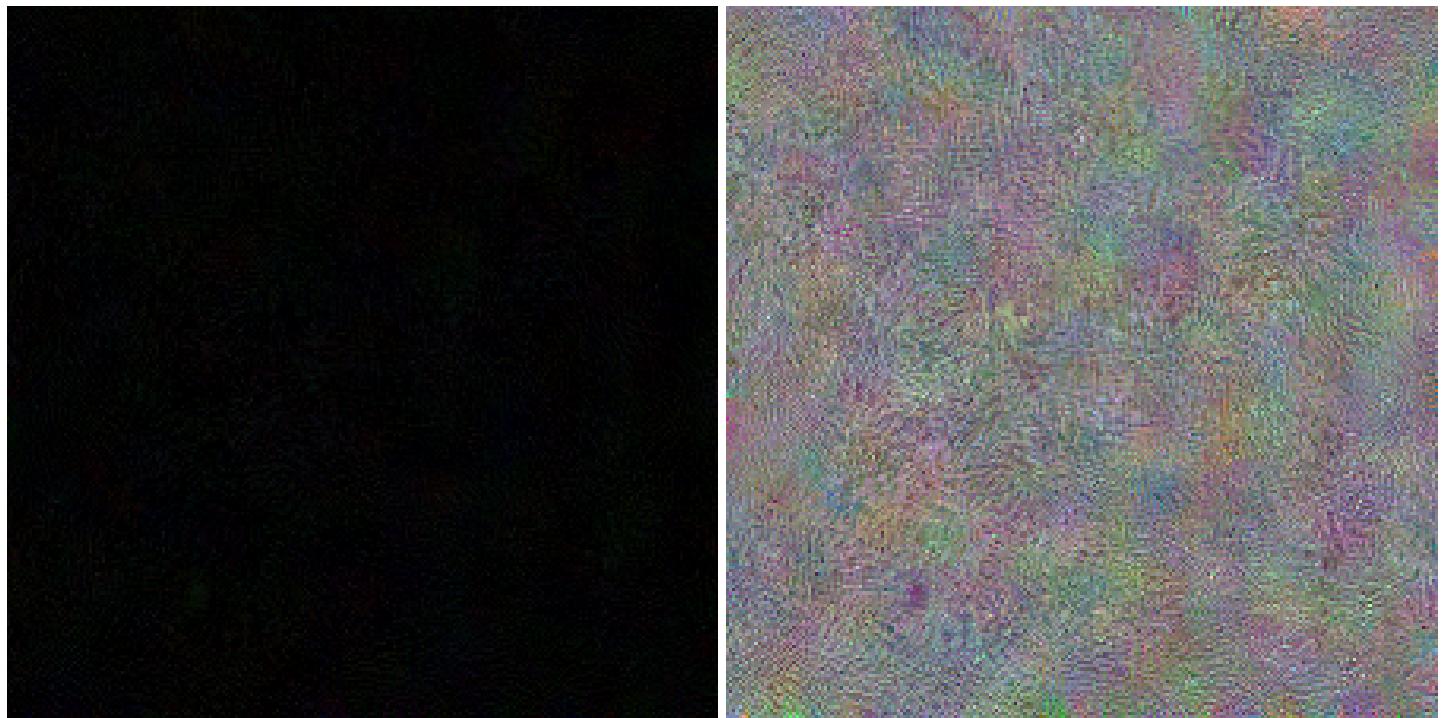


FIGURE 2 – Fond noir uniforme débruité à gauche et sa différence avec l'image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes dans aucune étape de l'algorithme.

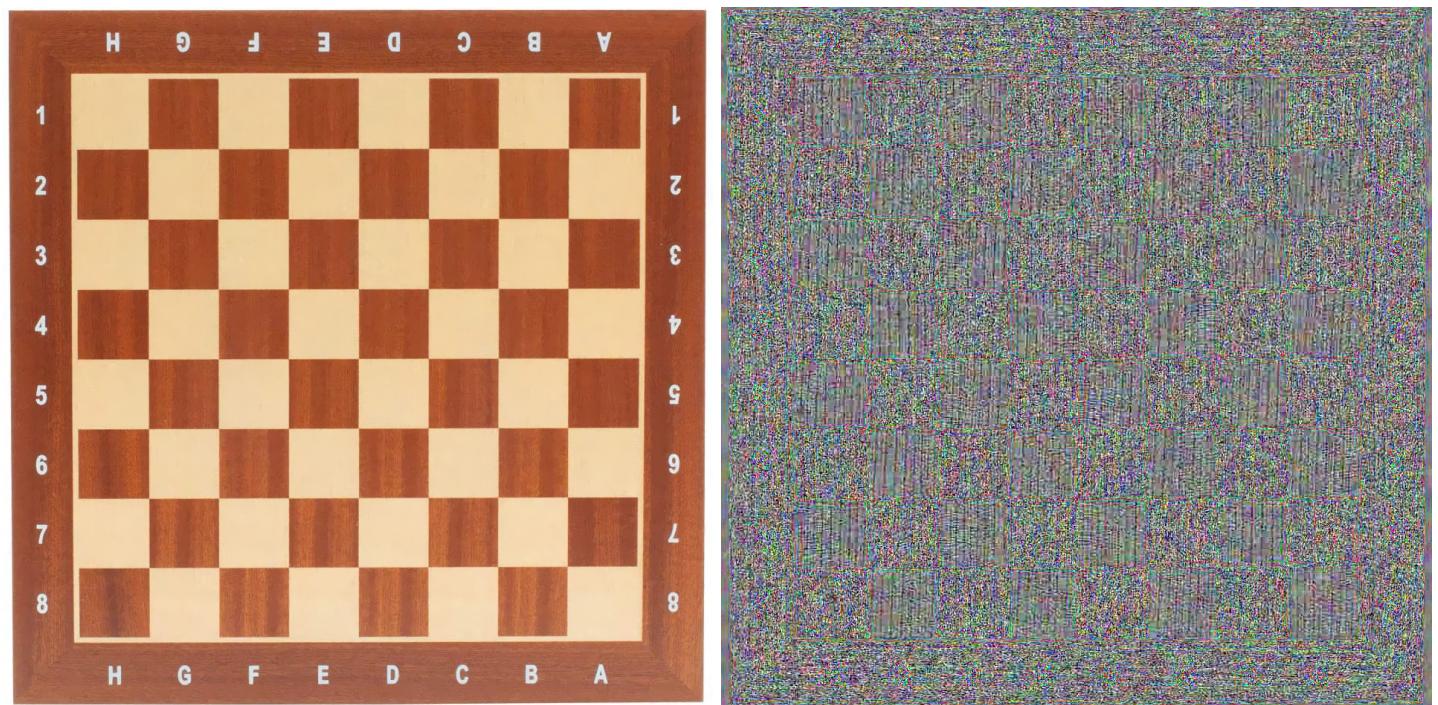


FIGURE 3 – Echiquier débruité à gauche et sa différence avec l'image bruitée à droite, pour  $\sigma = 10$ .

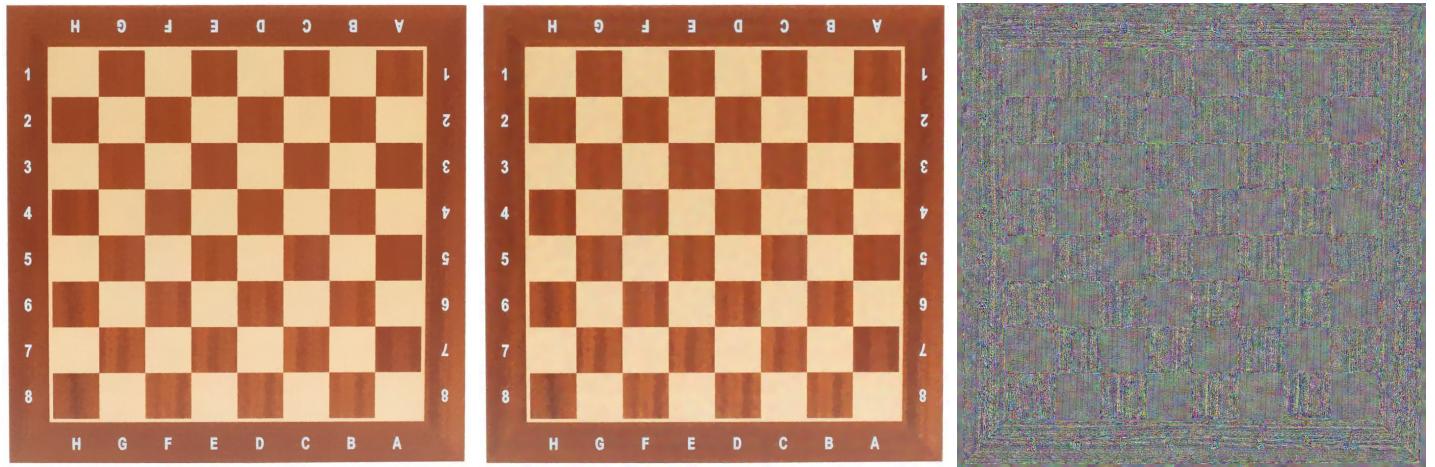


FIGURE 4 – Echiquier débruité après la 1<sup>ere</sup> étape à gauche puis le résultat final au centre et sa différence avec l'image bruitée à droite, pour  $\sigma = 30$ .

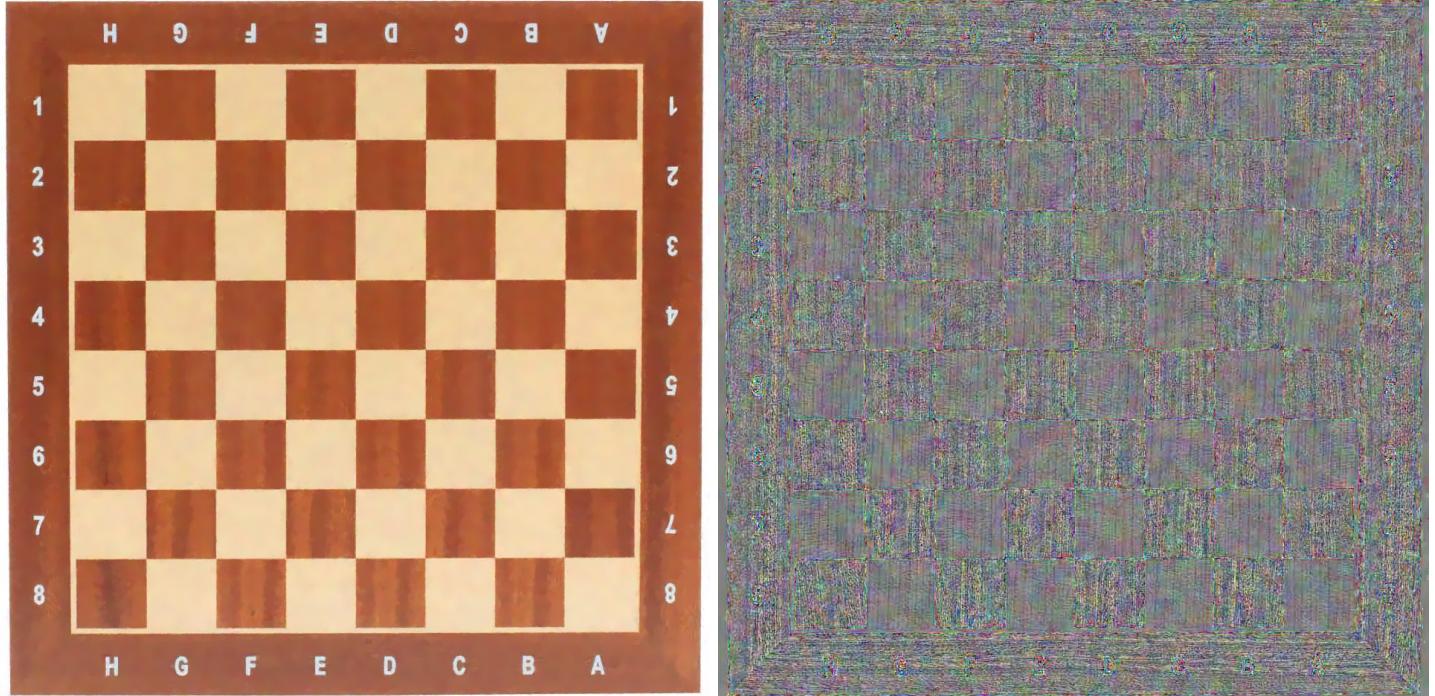


FIGURE 5 – Echiquier débruité à gauche et sa différence avec l'image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes pour les deux étapes de l'algorithme.

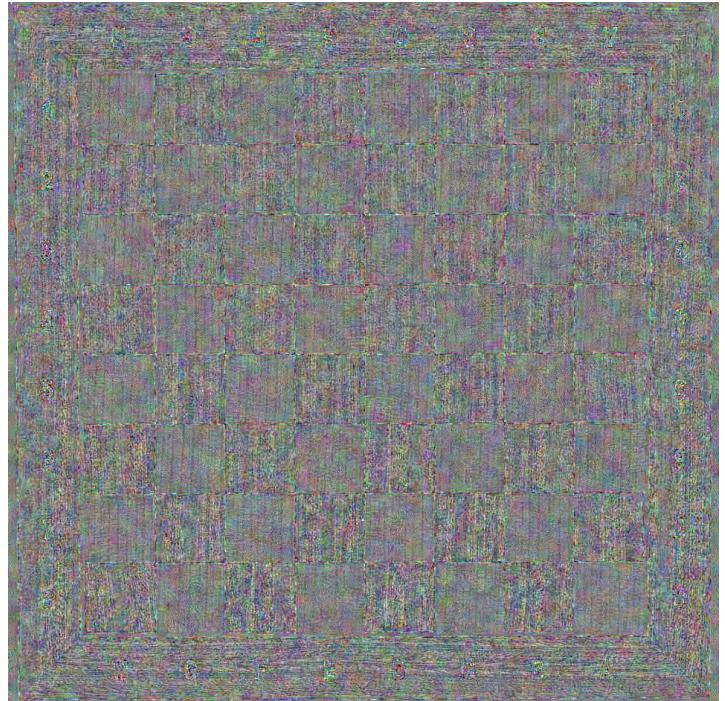


FIGURE 6 – Echiquier débruité à gauche et sa différence avec l'image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes dans aucune étape de l'algorithme.

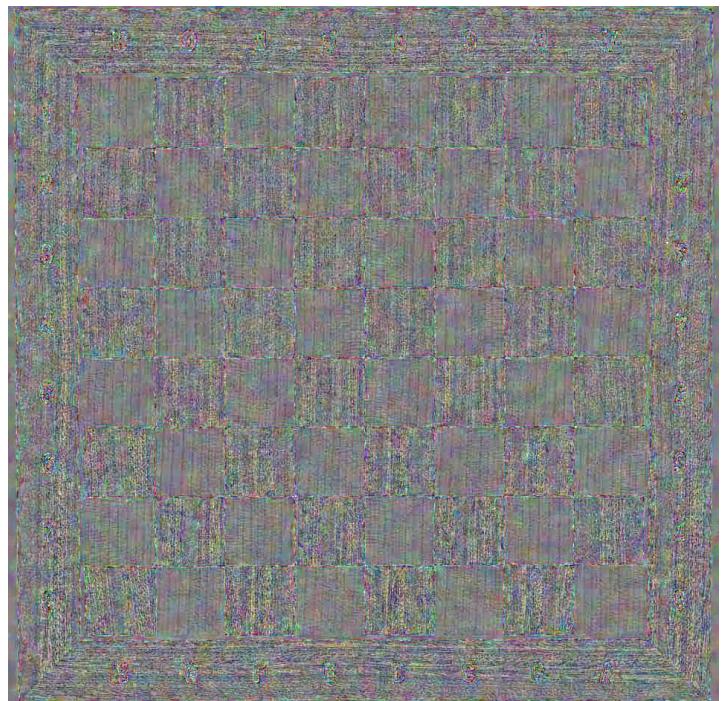
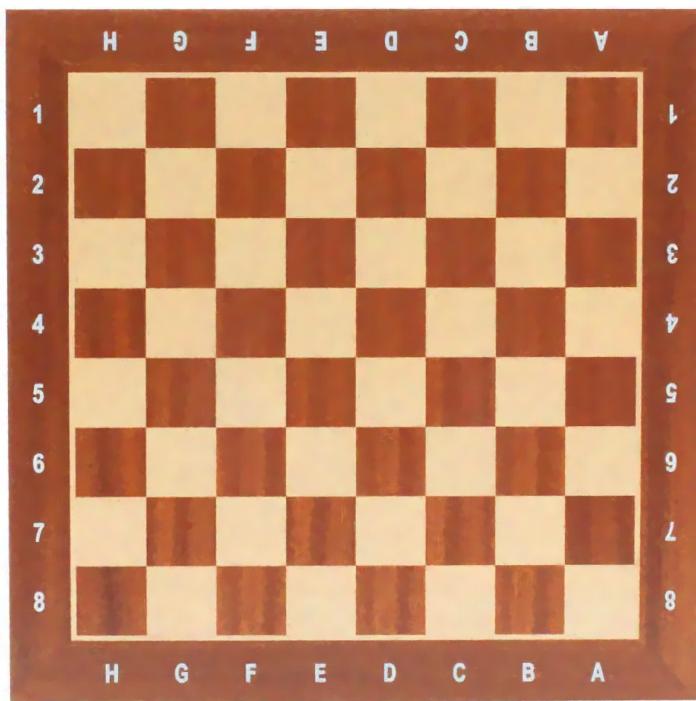


FIGURE 7 – Echiquier débruité à gauche et sa différence avec l'image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes dans la première étape de l'algorithme et calculé l'image de biais.

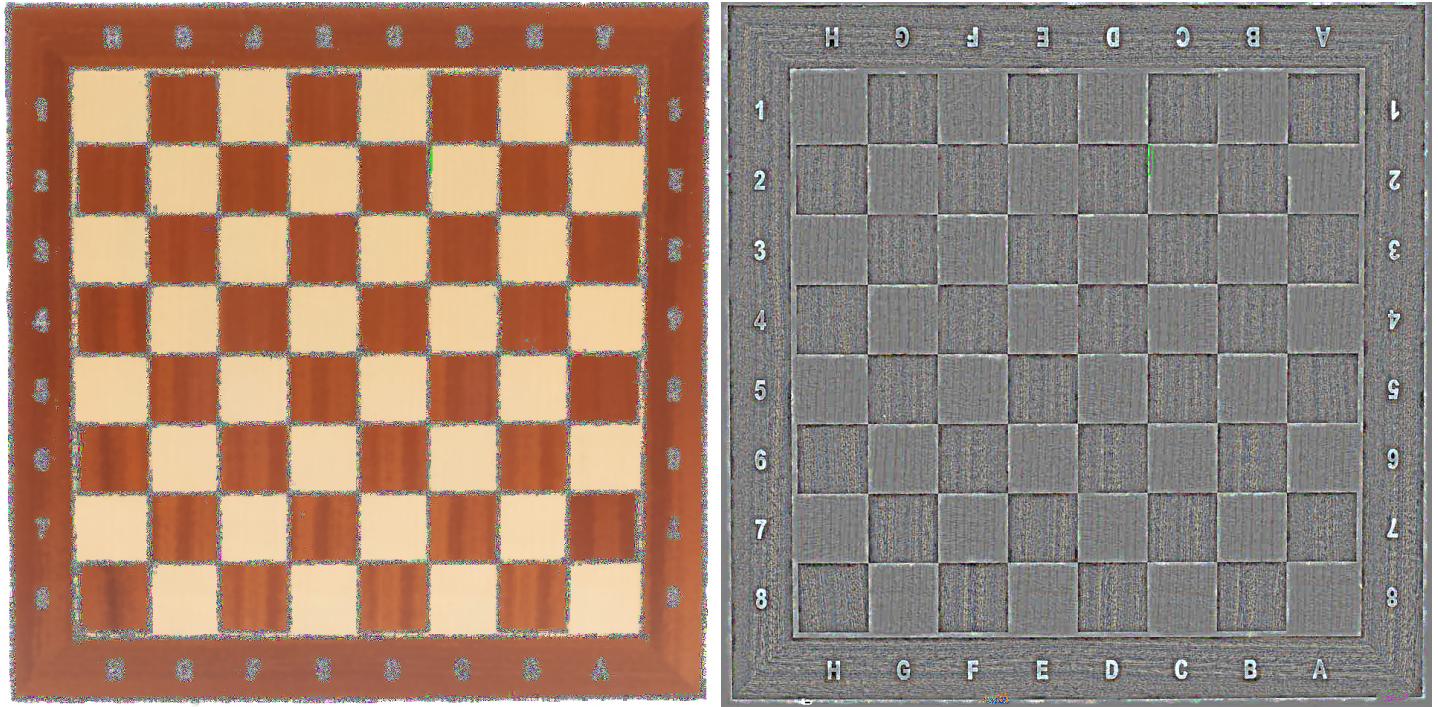


FIGURE 8 – Image de biais à gauche et différence de biais à droite, pour  $\sigma = 30$ .



FIGURE 9 – Echiquier débruité après la 1<sup>ere</sup> étape à gauche puis résultat final au centre et sa différence avec l'image bruitée à droite, pour  $\sigma = 60$ .

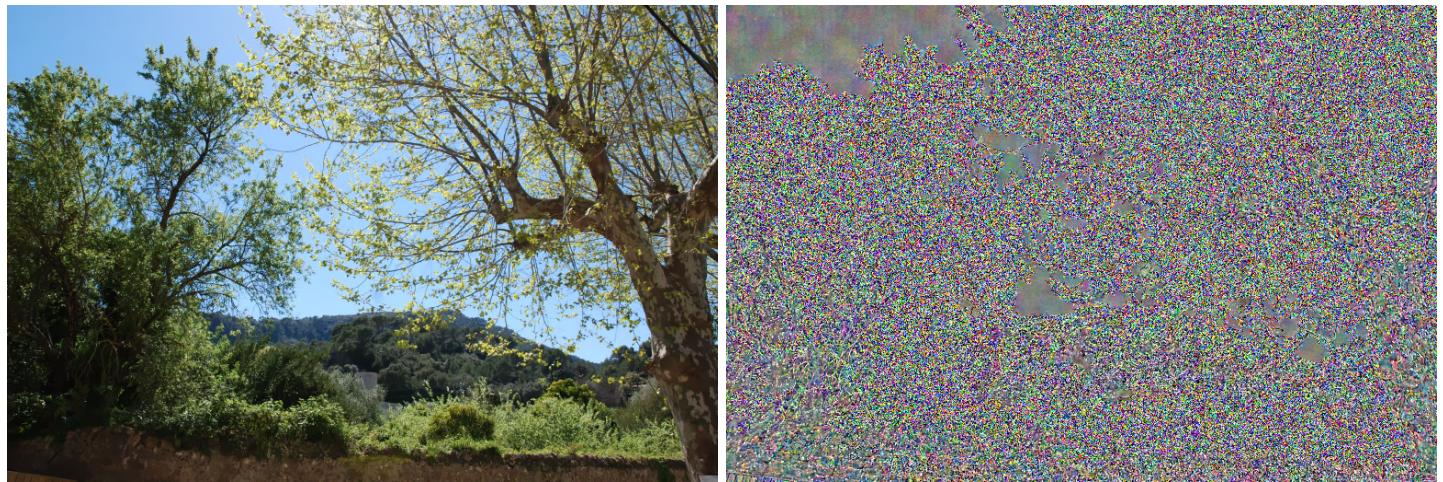


FIGURE 10 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour  $\sigma = 10$ .



FIGURE 11 – Photographie d’arbres débruitée après la 1<sup>ere</sup> étape à gauche puis le résultat final au centre et sa différence avec l’image bruitée à droite, pour  $\sigma = 30$ .

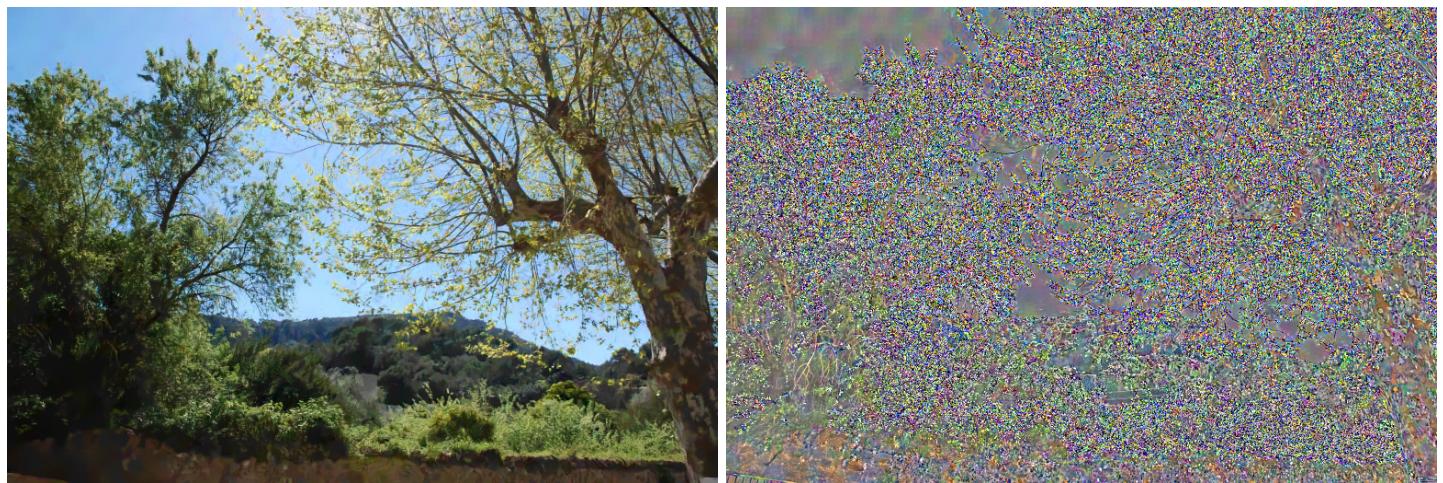


FIGURE 12 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes pour les deux étapes de l’algorithme.

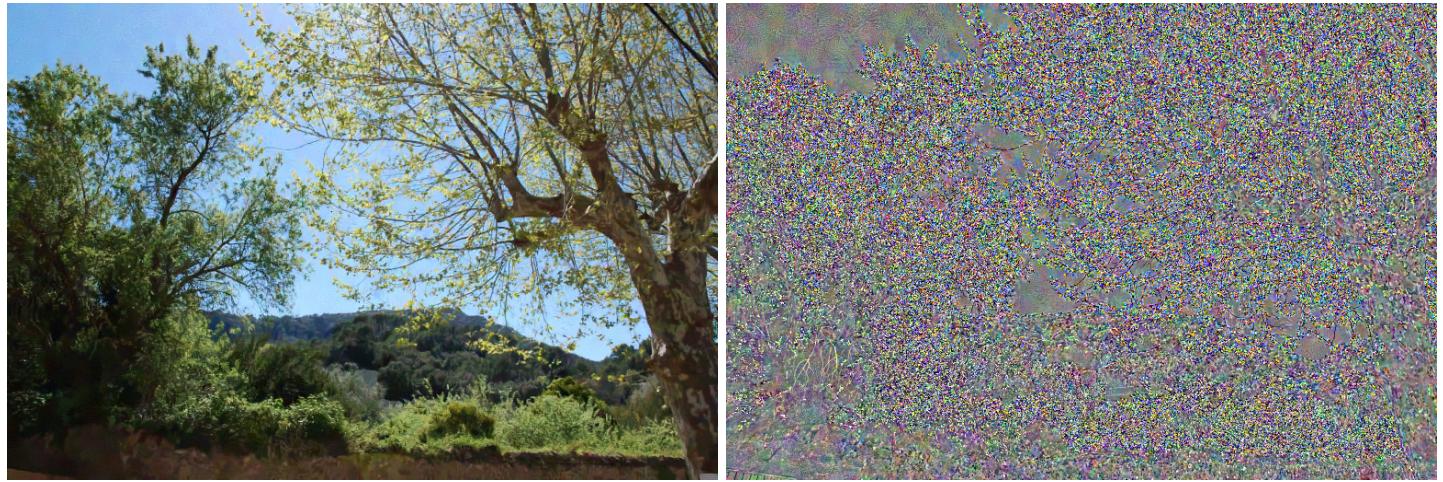


FIGURE 13 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes dans aucune étape de l’algorithme.

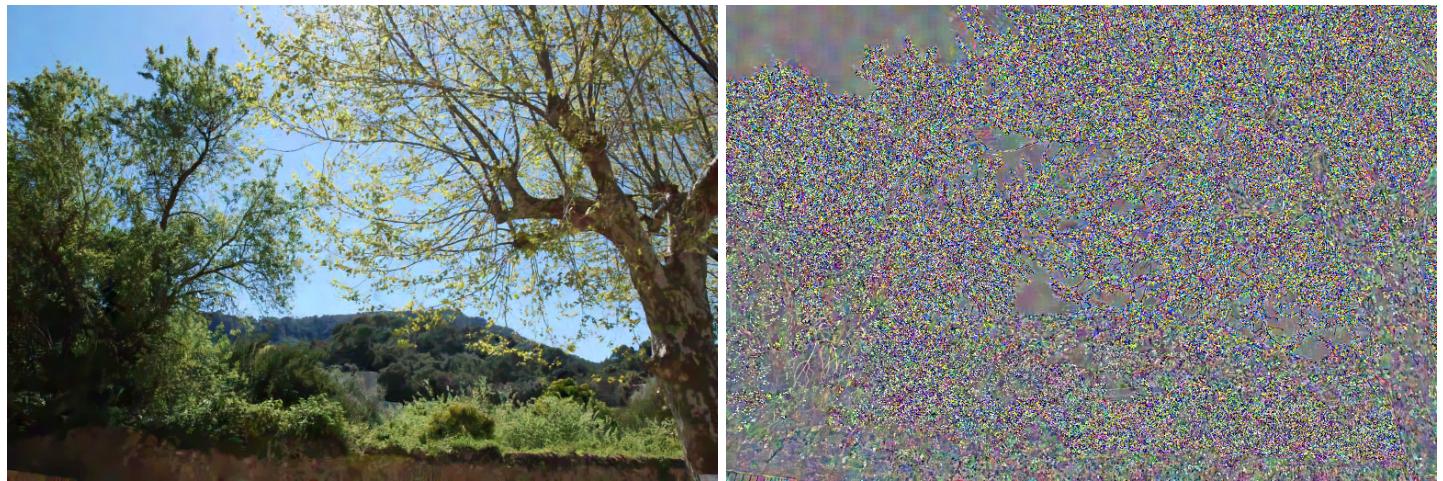


FIGURE 14 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour  $\sigma = 30$ . On a ici utilisé le critère de zones homogènes dans la première étape de l’algorithme et calculé l’image de biais.

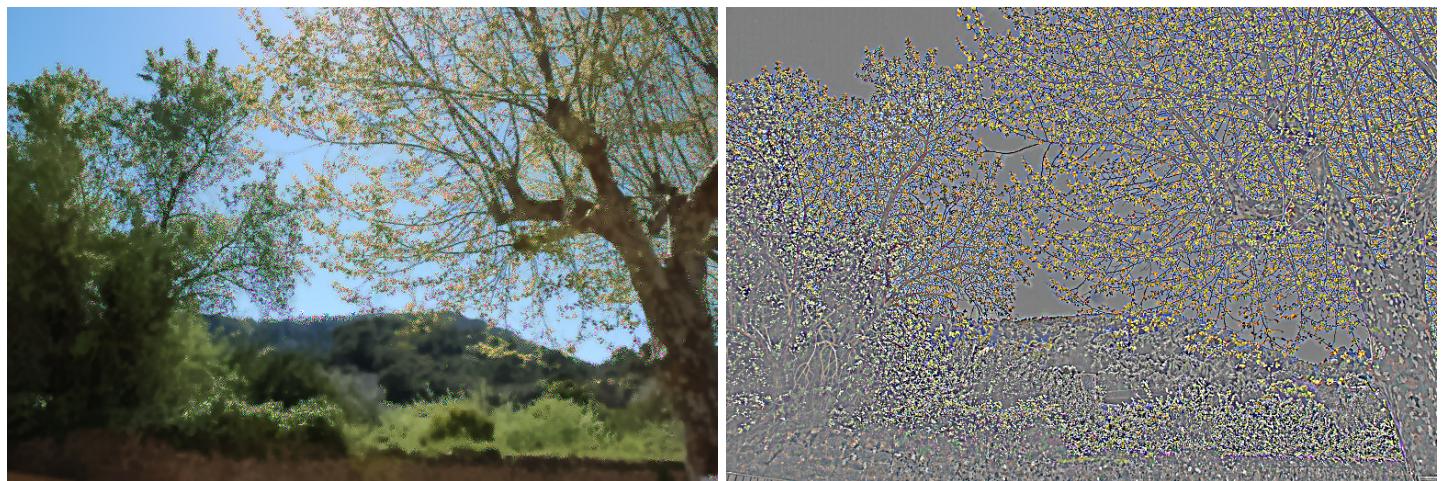


FIGURE 15 – Image de biais à gauche et différence de biais à droite, pour  $\sigma = 30$ .



FIGURE 16 – Photographie d’arbres débruitée à gauche et sa différence avec l’image bruitée à droite, pour  $\sigma = 60$ .

Image	$\sigma$	H.A.C. in 1 <sup>st</sup> step	H.A.C. in 2 <sup>nd</sup> step	Bias image computed	Time (s)	PSNR
Fond noir uniforme	10	Yes	No	No	0.56	51.3
	30	Yes	No	No	1.36	43.1
	30	Yes	Yes	No	1.29	49.7
	30	No	No	No	2.27	34.4
	30	Yes	No	Yes	1.36	43.3
	60	Yes	No	No	9.38	45.4
Echecs	10	Yes	No	No	5.09	37
	30	Yes	No	No	6.83	33.1
	30	Yes	Yes	No	5.68	33.1
	30	No	No	No	6.88	32.1
	30	Yes	No	Yes	5.98	33.1
	60	Yes	No	No	17.42	31.6
Arbre	10	Yes	No	No	1.97	31.7
	30	Yes	No	No	2.98	25
	30	Yes	Yes	No	2.98	25
	30	No	No	No	3.27	25
	30	Yes	No	Yes	3.07	25
	60	Yes	No	No	11.28	21.8

TABLE 1 – Temps d’exécution et PSNR obtenus par cet algorithme pour chaque image en fonction de la variance  $\sigma$  du bruit initial et des paramètres d’exécution de l’algorithme. Ici, *H.A.C.* signifie Critère de Zones Homogènes.

des changements brutaux de textures). En résumé, le PSNR est plutôt bon (supérieur à 32-33) mais le résultat visuel est assez mauvais et le temps de calcul est très variable selon les paramètres adoptés.

Pour la photo de paysage, lorsque le bruit est faible, le débruitage est excellent et semble même meilleur que celui de BM3D. On n'observe pas de grande différence entre la première et la deuxième étape du débruitage, et la différence résultat-image d'origine montre clairement qu'un bruit gaussien a été retiré de l'image. Le PSNR est très bon (31dB) et aucun artefact n'est présent, ni dans la végétation ni dans le ciel. Pour un bruit intermédiaire ( $\sigma = 30$ ), le temps est constant (entre 3 et 3.2s) et le PSNR reste aux alentours de 25. Dans tous les cas, la différence d'images met en avant certaines structures géométriques de l'image, notamment les branches en bas de l'arbre de gauche : l'algorithme ne supprime plus uniquement le bruit mais également la structure interne de l'image (ce qui implique une perte de qualité). La seconde étape de l'algorithme semble améliorer surtout les zones sombres de l'image en les éclaircissant pour mettre en avant leurs détails. On observe de nombreux artefacts dans les images, notamment dans le ciel où du bruit est encore très présent et des aberrations chromatiques sont apparues. L'image a clairement perdu en netteté et les zones à détails ont été uniformisées. Le fait de ne jamais utiliser de critère d'homogénéisation détériore encore plus le résultat final (on ne distingue plus de branches sur le tronc de l'arbre de gauche et les détails du muret ont disparu) et crée dans le ciel un bruit qui s'apparente à du dithering. Enfin, lorsque l'on utilise une image de biais, des artefacts sont toujours présents dans le ciel mais moins visibles que sans utilisation de biais. Si on a perdu en détails sur le muret et le tronc de droite, on semble avoir gagné en précision sur les zones affectées par le biais, i.e. les feuilles les plus proches du ciel, qui sont désormais plus nettes. Il y a donc un compromis à trouver entre netteté des zones affectées par le biais et perte d'information ailleurs. Finalement, pour un bruit fort, les détails des textures ont été perdus et de nombreux artefacts lumineux au bord des feuilles sont apparus : le débruitage n'est encore bon. Pour cette photo, le PSNR est plus faible (entre 22 et 25dB) et le temps de calcul est faible pour un bruit pas trop élevé mais explose pour  $\sigma = 60$ .

## Comparaison avec d'autres algorithmes

Pour l'échiquier, cet algorithme récupère bien mieux la texture du bois que les méthodes *Non-Local Means Denoising* et *Fast Approximation of Bilateral Filter* et conserve plus de netteté dans l'image. Par rapport à *BM3D*, les résultats sont identiques pour  $\sigma = 10$ , mais plus d'artefacts sont visibles pour  $\sigma = 30$  et de l'information a été perdue avec notre algorithme. Pour  $\sigma = 60$ , le résultat actuel comporte bien moins de textures sur le bois que *BM3D*. Cet algorithme est donc légèrement moins bon que *BM3D*.

Pour l'image de paysage, le résultat actuel est bien meilleur que *Fast Approximation of Bilateral Filter*. Par rapport à *Non-Local Means Denoising*, le résultat actuel comporte globalement moins d'artefacts. Les détails des arbres avaient considérablement disparus avec *NL Means* et ont été récupérés maintenant. Concernant *BM3D*, les résultats sont équivalents pour  $\sigma = 10$ . Pour  $\sigma = 30$ , notre résultat restitue plus de détails sur l'arbre de droite et sur les feuilles mais détériore l'arbre de gauche et ses branches. Pour  $\sigma = 60$ , les aberrations chromatiques que *BM3D* créait ne sont plus présentes, mais d'autres éléments ont perdu en netteté : alors que *BM3D* était net dans certaines zones et flou dans d'autres, la méthode actuelle donne une image uniformément floue.

En résumé, cette méthode est bien meilleure que *Non-Local Means* et l'approximation du filtre bilatéral. Elle donne des résultats très proches de ceux de *BM3D*, même si certains de ses résultats sont légèrement moins bons que *BM3D*. En revanche, cet algorithme est bien plus rapide que *BM3D*, mettant en moyenne 3 fois moins de temps pour débruiter une image.

## 2 Zoran and Weiss Gaussian Mixture

Le document ZoranWeissGMM présente une décomposition d'un ensemble de patchs d'images en un modèle de mélange gaussien. Il a été choisi de décomposer ces patchs en 400 gaussiennes. Un tableau présente pour chaque gaussienne sa probabilité dans le mélange gaussien, et chaque gaussienne est ensuite décomposée en patchs de vecteurs propres élémentaires avec les valeurs propres associées.

On remarque d'abord que les gaussiennes peuvent être classées en deux catégories : celles qui représentent des figures géométriques et informations très générales sur une image (comme la présence de variation de luminosité entre le haut et le bas de l'image) et celles qui modélisent des détails très particuliers et qu'on ne retrouve que rarement (comme des trous ou des intensités lumineuses concentrées en des points spécifiques). Les premiers sont plus évidents à analyser tandis que les seconds sont plus complexes, mais une fois tous ces patchs combinés, leur diversité permet à la fois de renseigner sur la structure globale de l'image et d'apporter des informations sur ses détails.

On constate également que beaucoup de gaussiennes se ressemblent très fortement (c'est le cas des deux gaussiennes à plus forte probabilité) : le découpage en 400 gaussiennes n'est donc peut-être pas le plus pertinent et un plus petit nombre aurait sûrement pu permettre de simplifier la décomposition sans perdre d'information.

Aussi, on remarque que, souvent, lorsque de l'importance est donnée aux petites valeurs propres d'une gaussienne, c'est que cette gaussienne a une probabilité faible. On peut comprendre cela par le fait que cette gaussienne va représenter des détails d'une image, qui sont moins fréquents sur un ensemble d'images et auront donc une probabilité d'apparition plus faible.

Analysons maintenant quelques gaussiennes en particuliers.

La plus grande probabilité est associée aux gaussiennes 53 et 201 ( $p = 0.0395$ ). Leurs premiers vecteurs propres détectent des variations horizontales et verticales, comme des gradients, puis des diagonales et figures légèrement plus complexes. Ces gaussiennes détectent donc la structure plus globale de l'image. Toutes les valeurs propres de la gaussienne sont néanmoins élevées, ce qui permet d'approcher avec plus de précision des structures régulières et répétées. Les patchs simulés semblent d'ailleurs montrer des structures complexes mais plutôt géométriques.

La gaussienne 4 détecte les diagonales de l'angle haut-gauche à celui bas-droite. De nombreuses valeurs propres sont élevées afin de viser une plus grande variété de diagonales et bandes transversales. Cette structure est spécifique et pas présente dans toutes les images, ce qui explique la probabilité plutôt faible de la gaussienne. La gaussienne 21 semble faire la même chose mais cette fois pour des bandes verticales.

La gaussienne 12 semble faire ressortir les trous dans les images. Celle n°16 montre plutôt des quadrillages sur le bas à droite dans les images, quadrillages légèrement tourné. La probabilité de cette gaussienne est également faible puisque le patch est spécifique.

La gaussienne 136 semble générer des patchs de bâtiments ou photos prises dans la rue, avec la chaussée, des arbres et des voitures sur les derniers patchs simulés et des bâtiments clairs sur les premiers patchs simulés.

Enfin, la gaussienne 127 a elle aussi des vecteurs propres très généraux. Pourtant, les patchs simulés de cette gaussienne semblent faire apparaître des visages humains, figures animales ou casques.

Tous ces patchs semblent donc être représentatifs d'une diversité d'éléments généralement présents dans des photographies, ce qui les rend complémentaires.

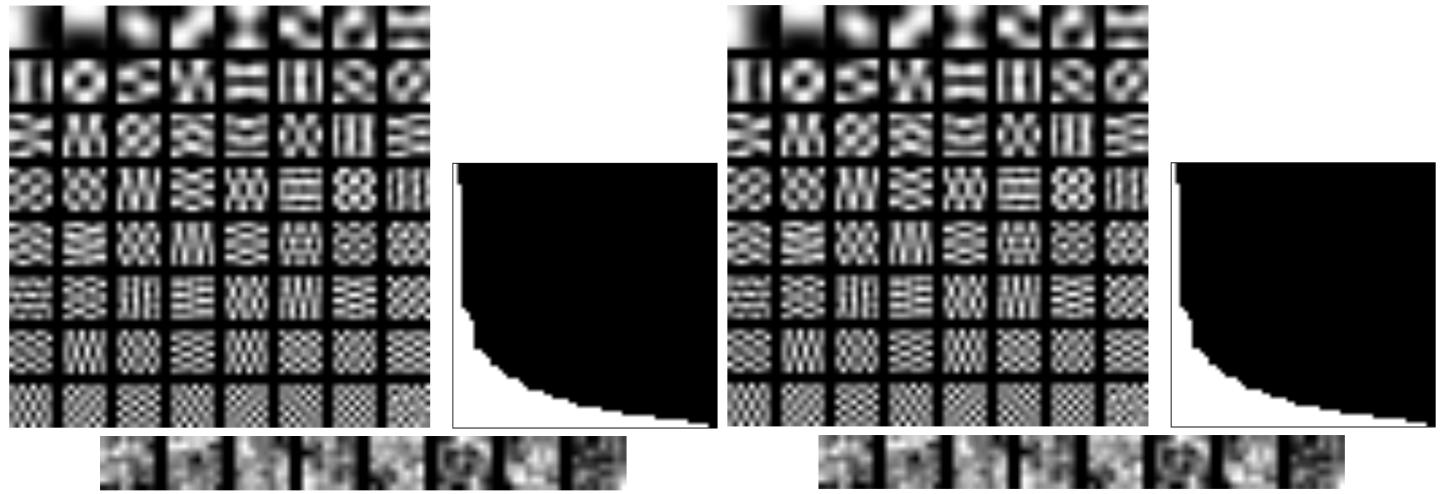


FIGURE 17 – Gaussiennes 53 à gauche et 201 à droite. Ce sont les gaussiennes de plus forte probabilité dans ce GMM ( $p = 0.0395$ ).

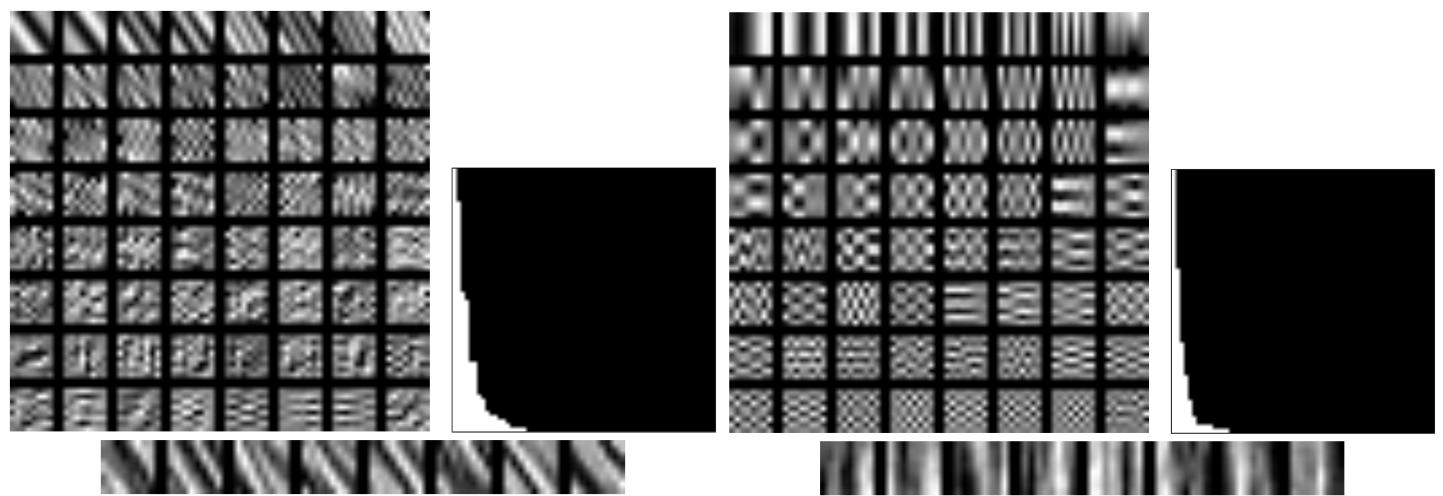


FIGURE 18 – Gaussiennes 4 à gauche et 21 à droite. Elles détectent des structures géométriques très précises. Probabilités :  $p_4 = 0.00072$ ,  $p_{21} = 0.00472$ .

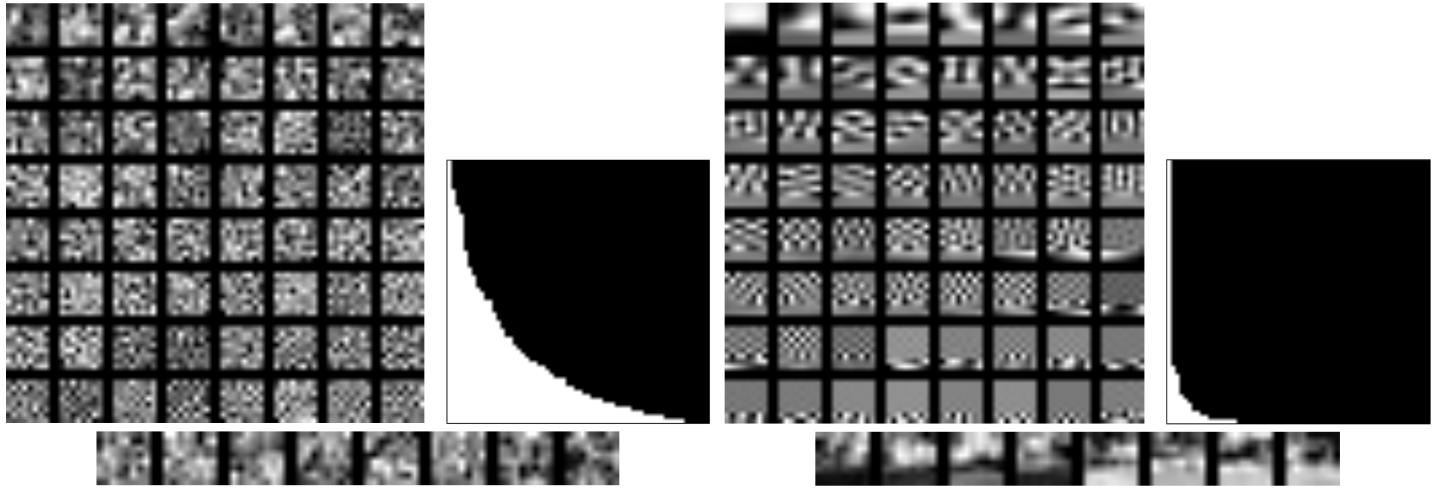


FIGURE 19 – Gaussiennes 12 à gauche et 136 à droite. La gaussienne à gauche semble détecter des trous et structures très complexes. Celle de droite semble détecter des bâtiments (4 premiers patchs simulés) et des images de rue (4 derniers patchs simulés). Probabilités :  $p_{12} = 0.00010$ ,  $p_{136} = 0.00174$ .



FIGURE 20 – Gaussiennes 127, de probabilité  $p = 0.02723$ . Elle semble détecter des visages : le premier patch simulé ressemble à un homme avec une casquette, les 3 suivants à des têtes de chats (avec des rotations de 90°), et le dernier ressemble à un casque de Boba Fett.