

- **Quelle méthode TVdeconv utilise-t-elle pour résoudre les problèmes de bords?**

Il y en a 3 possibilités de gestion des effets de bord:

Si edgehandle= 'taper' alors on ajoute à l'image une bordure lisse ;
Si edgehandle= 'sym' alors on symétrise l'image ;
Si edgehandle= 'nothing' alors on ne fait rien (mauvais).

Par défaut, TVdeconv utilise la gestion “taper”, qui consiste à flouter une image I par le noyau K (circulairement) donnant une image J et mélanger l'image I avec cette image J de façon que J soit prépondérante aux bords. Comme ça, l'image J déconvolué par le noyau K n'aura pas d'effets de bord.

- **Quelle est la solution Wiener du même problème de déconvolution (IMA201 et IMA203)?**

La solution par le filtre de Wiener résulte en un compromis entre un filtrage inverse et le lissage de l'image (suppression du bruit).

$$\hat{f}(\omega) = \frac{\overline{\hat{K}(\omega)}}{|\hat{K}(\omega)|^2 + \frac{\sigma_b^2}{\sigma_s^2(\omega)}} \hat{g}(\omega)$$

ω parcourt les fréquences de Fourier
 $\sigma_s^2(\omega)$ puiss. du signal à la fréq. ω

- **TVdeconv est-elle meilleure que Wiener?**

Non, les deux méthodes ont ses avantages et désavantages. La méthode de Variation Totale, par exemple, peut rendre lisse des zones avec une texture non-lisse (puisque les textures augmentent la variation totale). Pourtant, les images déconvolués par TVdeconv sont en général moins floues.

- **SinglePhaseRetrieval est stable au bruit, mais jusqu'à quel point?**

J'ai essayé plusieurs valeurs de bruit. Jusqu'à 7 fois la valeur du bruit initialement codé dans le TP (`np.random.rand(Mh,Mh)-0.5`)*HKm/2), le SinglePhaseRetrieval est robuste, comme on peut voir ci-dessous. Si la valeur du bruit est plus grande que ça, on n'arrive pas à récupérer le noyau.

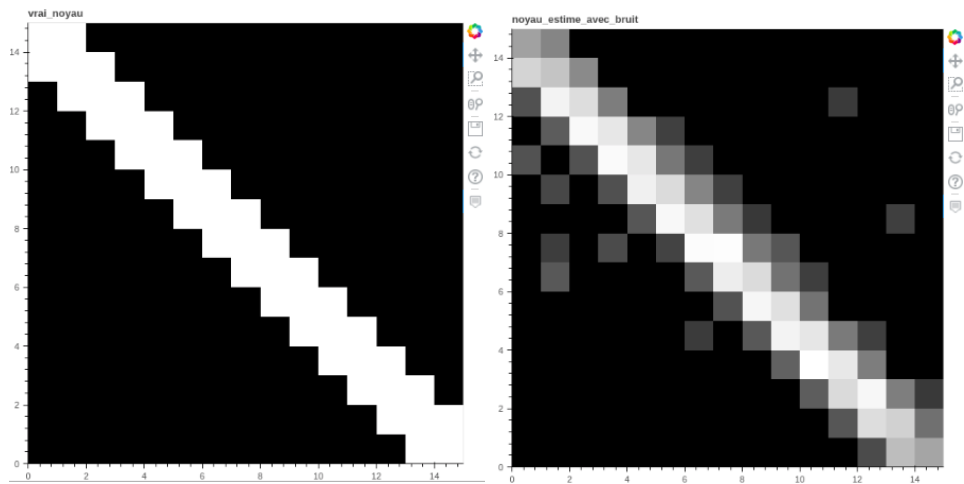


Figure 1 - Vrai noyau vs. Noyau estimé avec bruit d'écart type 7x HKm/2

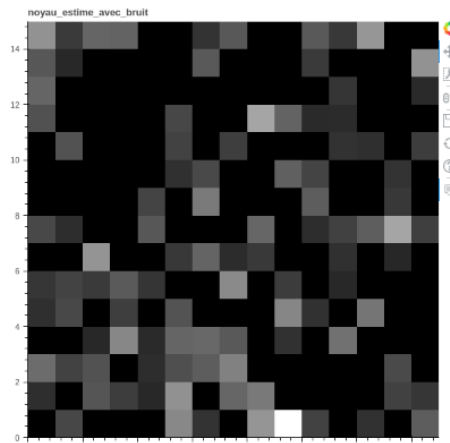
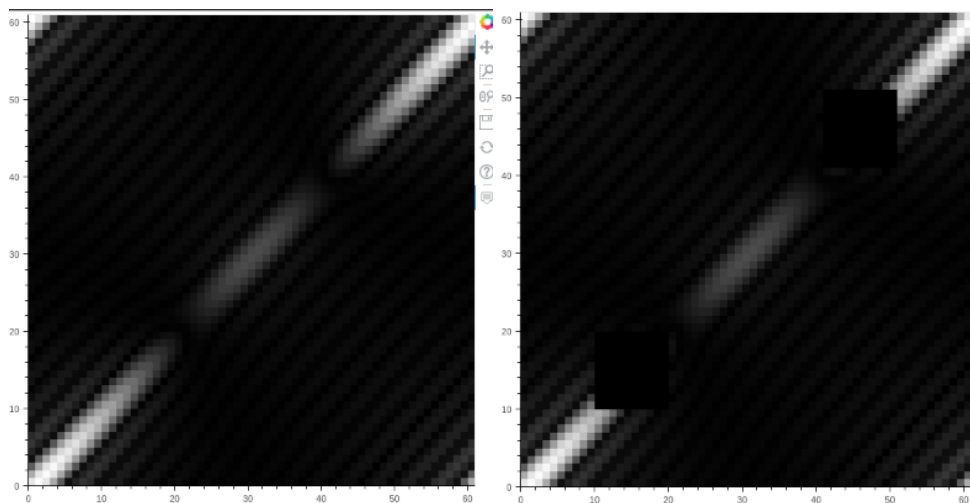
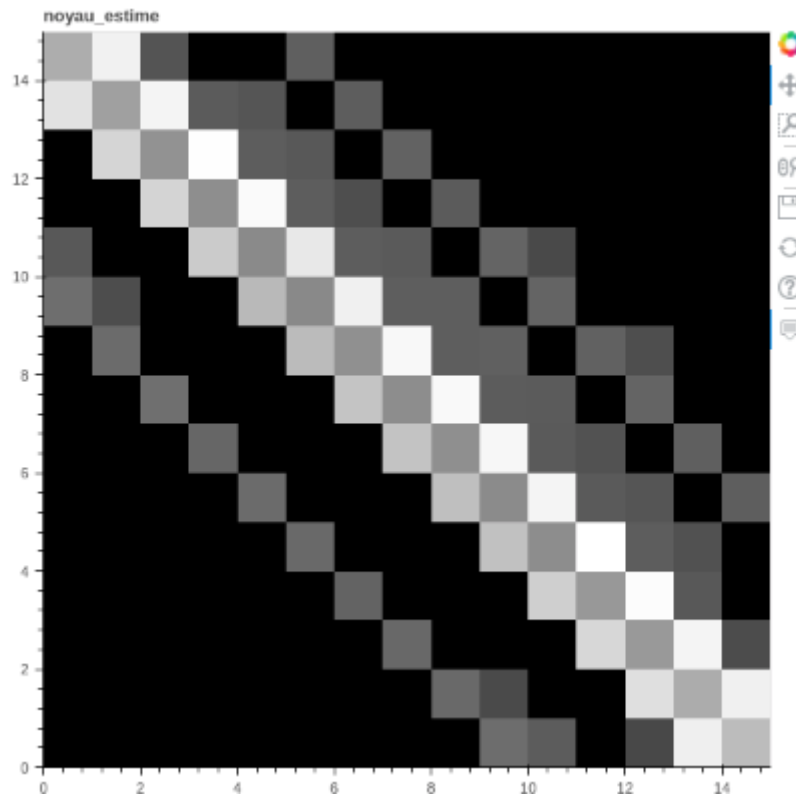


Figure 2 - Noyau estimé avec bruit d'écart type 10x HKm/2

- Si on introduit une discontinuité dans le spectre, que cela donne-t-il sur le noyau?



J'ai introduit deux carrés noirs le long de la diagonale principale du spectre, comme on peut voir ci-dessus. On peut voir sur l'image suivante qu'en gros le noyau est toujours trouvé, mais avec un sort de *ringing* (il y a des lignes parallèles à la diagonale principale du noyau).



- Démontrer que l'autocorrélation de la projection est la même chose que la projection de l'autocorrélation.
- La méthode originale propose de calculer pour chaque θ la projection du gradient θ puis autocorrélation. Dans le script fourni qu'est-ce qui est fait? Pour quelles tailles d'images cela devient-il intéressant.

Le script fourni utilise la propriété de la question précédente, "l'autocorrélation de la projection est la même chose que la projection de l'autocorrélation", dans la fonction `calcul_correlations_initiales`

- Pourquoi y-a-t-il des /255 dans le code (les auteurs donnent des lambda pour des images entre 0 et 1)? Justifier en vous référant aux fonctionnelles à minimiser.
- En visualisant des images Cb et Cr, pourquoi ne sont-elles pas défloutées à votre avis?

On voit la différence ci-dessous les images Cb et Cr. Pour obtenir une image couleur, la défloutage est fait sur le canal plus important spatialement, puis les autres canaux sont ajoutés pour donner les informations de couleur.

