TP flou IMA206

BLANCHARD Théau

Déconvolution par variation totale.

Pour éviter les effets de bords, *deconvTV* va mélanger l'image floutée avec une nouvelle image que l'on va flouter circulairement selon le noyau choisi de sorte que la deuxième soit prépondérante sur les bords de l'image. Cela fait en sorte que lorsque l'on va déconvoluer, son influence va être négligeable sur les bords et ainsi éviter de créer des artefacts. On peut aussi choisir de symétriser l'image ou de ne rien faire mais les résultats sont moins probants.

Plus le terme d'attache aux données est important, plus le résultat sera proche de l'image originale, c'est-à-dire bruité. En revanche, plus il est faible, plus on va minimiser la variation totale, ce qui est réalisé pour une image uniforme. Donc plus le terme le paramètre d'attache aux données est faible, plus le résultat est flou.

La solution Wiener de ce problème serait :

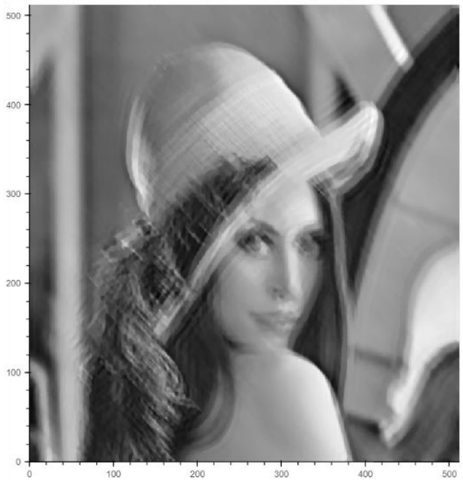
$$\hat{f}(\omega) = \hat{g}(\omega) \frac{\overline{\widehat{K}(\omega)}}{|\widehat{K}(\omega)|^2 + \frac{{\sigma_b}^2(\omega)}{\sigma_s(\omega)^2}}$$

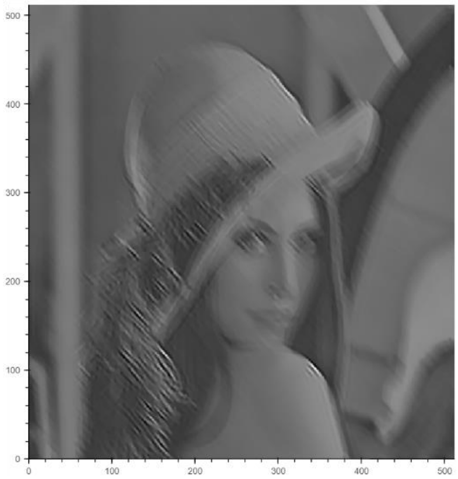
Contrairement à la méthode de Weiner, cette méthode ne nécessite pas une connaissance aussi fine du noyau. C'est pour cela que le filtre de Wiener est plus adapté pour du débruitage que de la déconvolution.

Évidemment plus le bruit est important plus il est difficile de déconvoluer l'image sans avoir en contrepartie un bruit très important.

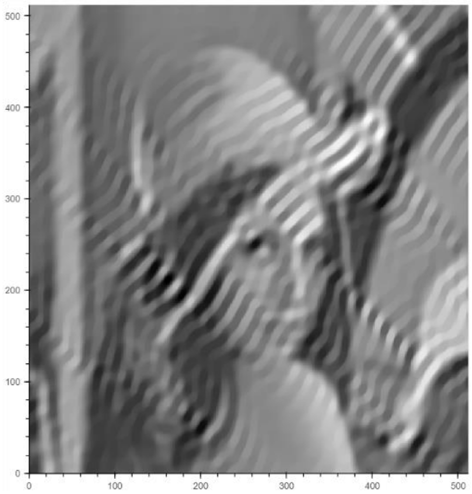
En revanche si on se trompe de noyau, même au niveau de la taille on aura des résultats assez vite mauvais. Avec une fenêtre constante de 15x15, on obtient l'image ci-contre, qui lorsque l'on déconvolue avec le même noyau donne :

Mais donne si on déconvolue avec un noyau constant 7x7, ou un noyau aléatoire 7x7 :



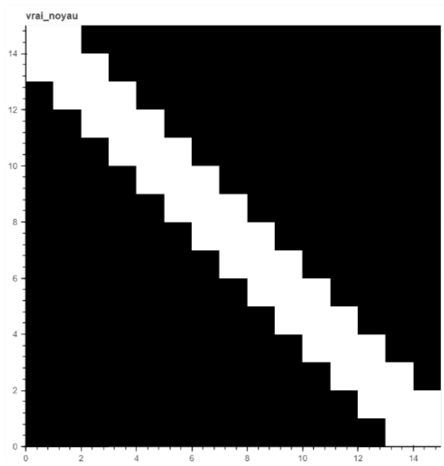


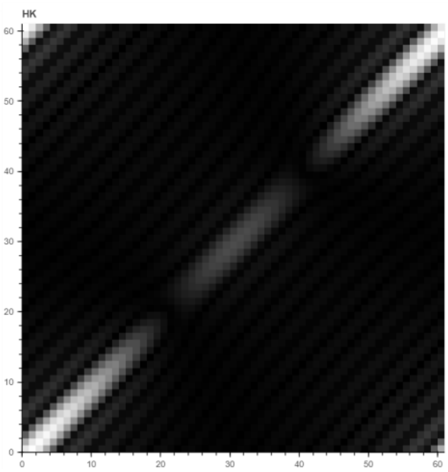
Pire encore si on essaie de déconvoluer un noyau gaussien avec un noyau constant :



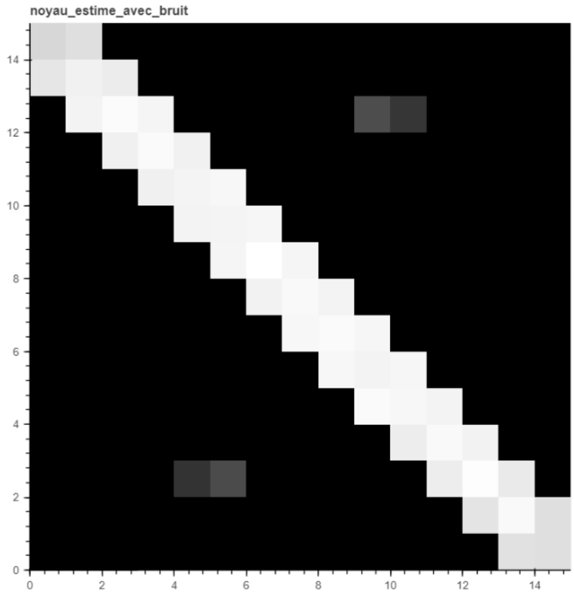
Single Phase Retrieval :

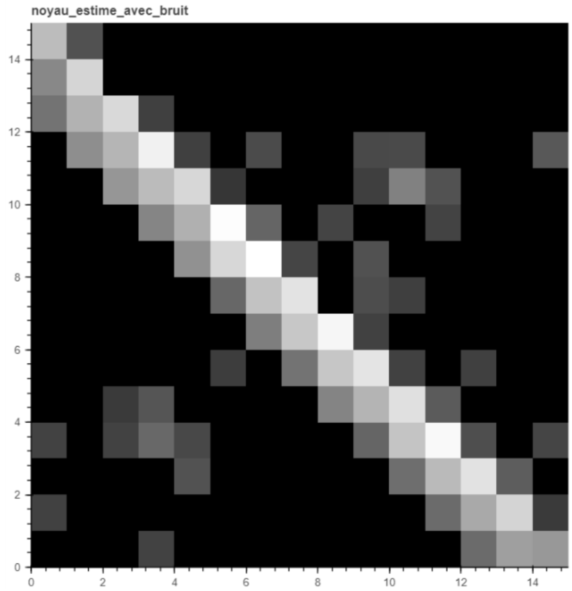
On voit lorsque qu'il n'y a pas de bruit on obtient exactement le bon résultat. Ci-contre le noyau vrai et l'amplitude de sa TF.



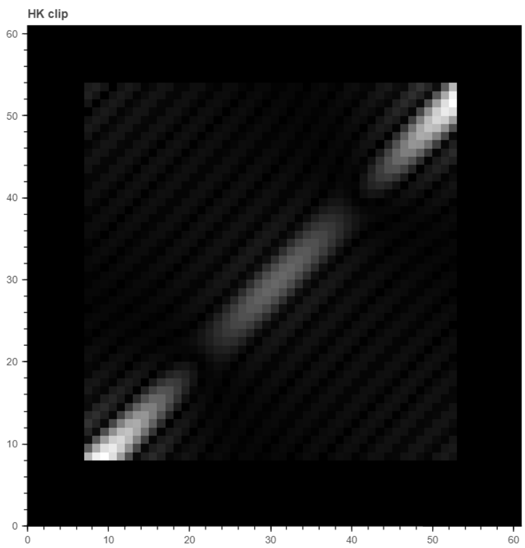


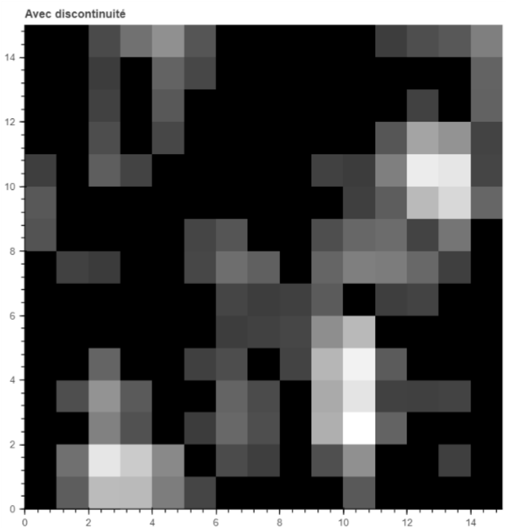
On voit que la méthode est résistante au bruit jusque dans une certaine mesure. Si on ajoute un tirage aléatoire entre -0.5 et +0.5 multiplié par k fois la moyenne du spectre, on constate que jusqu'à k = 1 on obtient toujours de bons résultats même s'il y a quelques erreurs, par contre au-delà les résultats sont moins probants (exemple avec k=4).





Enfin si on ajoute une discontinuité dans le spectre, en clipant par exemple les hautes fréquences, les résultats sont de suite très mauvais. On aurait des résultats similaires en introduisant des discontinuités directement sur le noyau.





Autocorrélation et Projection

Méthode entière

Dans le script on effectue un shear sur l'image avant de projeter. Cela revient en fait bien à projeter l'entrée sur toutes les droites d'angles theta directement. Cette méthode devient intéressante pour des images assez grandes où il serait vite couteux de réaliser vraiment la projection orthogonale de *chaque* point de l'image sur *chaque* droite.

Le fait de remettre l'image entre 0 et 1 en divisant par 255 permet d'utiliser des valeurs de lambda qui sont normalisés de sorte qu'il n'y ait pas à pondérer le terme de régularisation dans la fonctionnelle. Dans la pratique cela permet de conserver les valeurs d'hyperparamètres utilisés dans l'article.

Il n'y a pas besoin de déflouter les images Cr et Cb de l'espace YCrCb car l'œil humain étant beaucoup plus sensible à la chrominance qu'à la luminance, il suffit de travailler sur Y pour obtenir des résultats largement satisfaisant.