Mona Mokart

# TP IMA206 : Le flou

## I - Variation Totale :

1. Quelle méthode TVdeconv utilise-t-elle pour résoudre les problèmes de bords ? En particulier comment fonctionne la gestion « taper » ? Comment le bruit influe-t-il sur le résultat ?

Il y a deux solutions possibles pour résoudre les problèmes de bords. La première est de symétriser l'image initiale et d'appliquer la déconvolution sur celle-ci puis récupérer l'image :

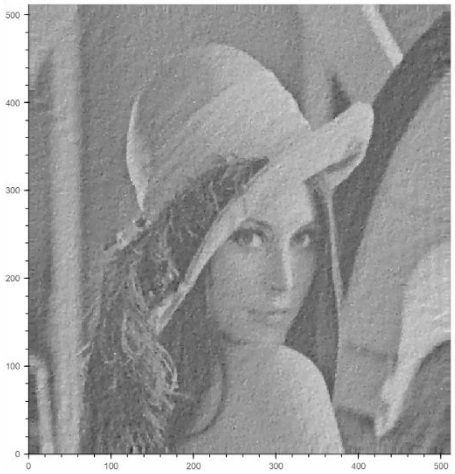

La deuxième (« taper ») solution est de rajouter des bandes d'une certaine largeur autour de l'image, qui prennent la valeur des bords puis d'y appliquer un filtre de flou de noyaux K. Ensuite l'image résultante J est superposée avec l'image « padée » sans filtre de façon que l'image J soit prépondérante sur l'image finale au niveau des bords. De cette façon on évite les effets de bords et après la déconvolution on recadre l'image dans sa taille initiale :

Influence du bruit :

Sans bruit :

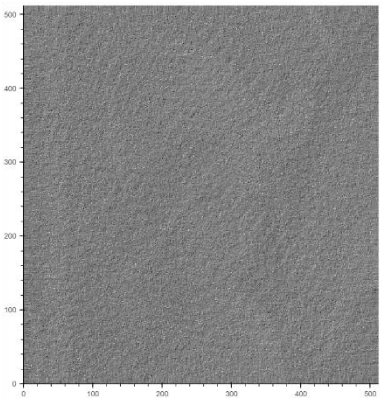
Floutée déconv

Avec un bruit léger :



Floutée + bruitée (σ=5) déconv

Avec un gros bruit :



Floutée + bruitée(σ=25) déconv

On voit ici que si on ne change pas le paramètre de régularisation sur la TV, en augmentant le flou, la déconvolution ne suit pas car on n'impose pas assez le débruitage. En revanche si on change le paramètre lambda on arrive quand même à obtenir un résultat satisfaisant :


déconv de l'image Lena floutée + bruitée (σ=5)avec un autre paramètre lambda

Evidemment plus il y aura du bruit moins l'image déconvoluée sera proche de l'image parfaite mais on arrive quand même à récupérer une image acceptable.

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

La solution f est telle que :

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

3. TVdeconv est-elle meilleure que Wiener?

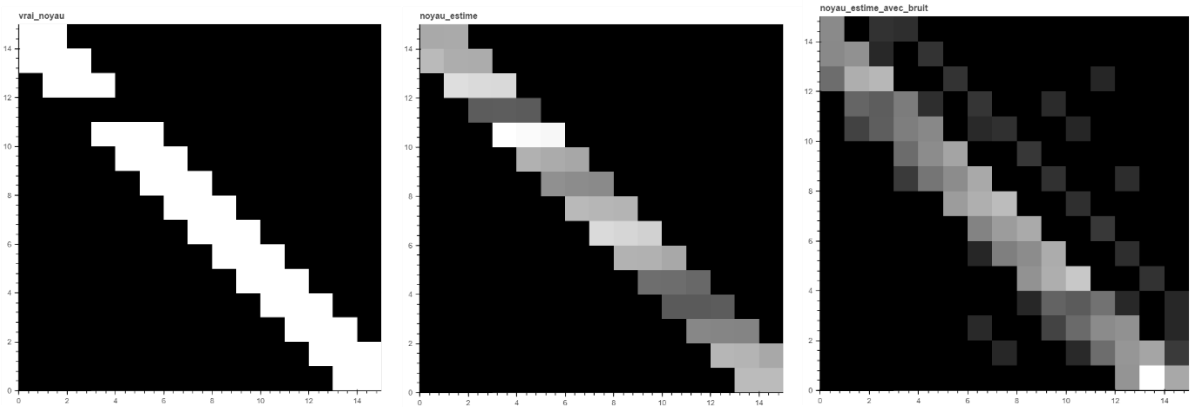
Wiener minimise comme contrainte l'énergie du gradient de f et non la variation totale, ce qui donne forcément un résultat différent. Ici par exemple avec un bruit gaussien de σ=5 on obtient ce résultat avec Wiener :

Je trouve que la méthode avec TV est meilleure que wiener car ici on observe des vaguelettes. Alors que l'énergie du gradient impose une image lisse, la contrainte TV impose des zones constantes par morceaux ce qui empêche d'obtenir ces vaguelettes.

Sur ce genre de flou orienté je pense qu'il est mieux d'utiliser la variation totale

## II – SinglePhaseRetrieval

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




Ici on voit qu'avec une simple discontinuité, l'estimation du noyau devient mauvaise et avec du bruit elle le devient encore plus.

## III – Méthode complète

1. Démontrer que l'autocorrélation de la projection est la même chose que la projection de l'autocorrélation.
2. 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.

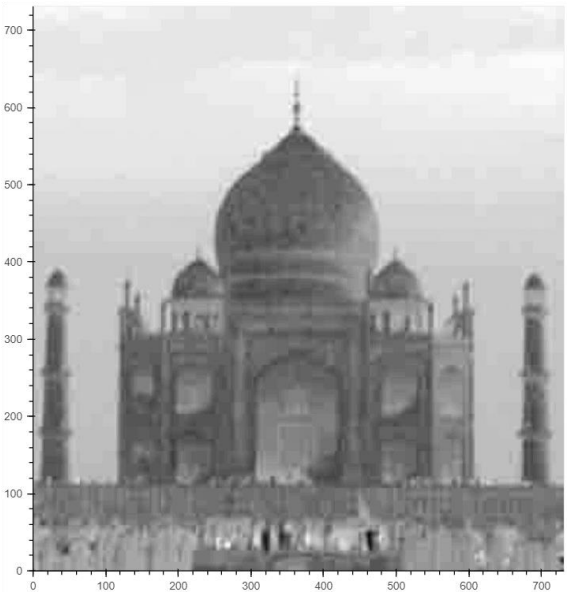
Au lieu de faire une projection directe, l'algorithme effectue un shear avant, puis projette, ce qui est équivalent à projeter avec un angle.

3. 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 référent aux fonctionnelles à minimiser.

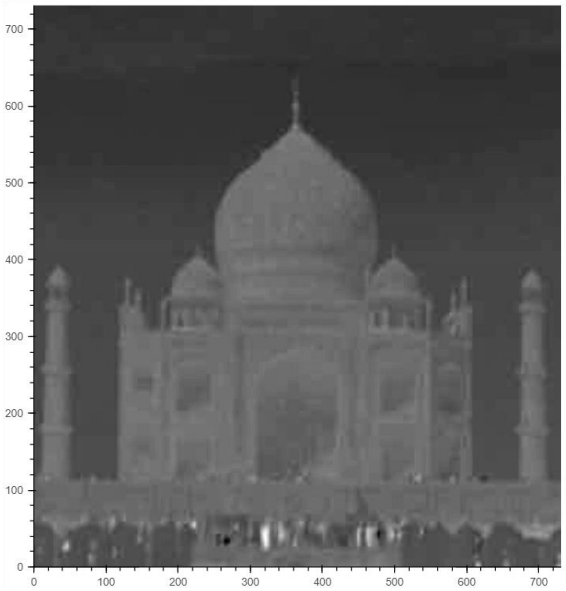
Ici on travaille sur des images de valeurs de pixels entre 0 et 255. Les coefficients gamma et lambda apparaissent dans la fonction à minimiser sur des normes au carrée de fonctions prenant des valeurs entre 0 et 255. Lors de l'étape de descente de gradient, on dérive cette norme au carré qui devient une fonction de degré 1 donc les valeurs prises par le gradient sont entre 0 et 255. Ainsi diviser par 255 les coefficients de régularisation sur nos images est équivalent à garder les bons coeff initiaux de l'article sur leurs images de pixels entre 0 et 1.

4. En visualisant des images Cb et Cr, pourquoi ne sont-elles pas défloutées à votre avis ?

Ici on travaille sur une image en couleur. Or il est compliqué d'appliquer un tel algorithme pour trois canaux. On peut supposer que le flou ne dépend pas de la couleur et qu'il est dû à l'appareil et au mouvement qui sont bien indépendants de la réception des couleurs. Ainsi ce qui serait pratique serait de travailler sur l'image en noir et blanc puis de réappliquer les couleurs. Or il faudrait qu'en réappliquant ces couleurs on ne créé pas un flou « gardé » par ces canaux.

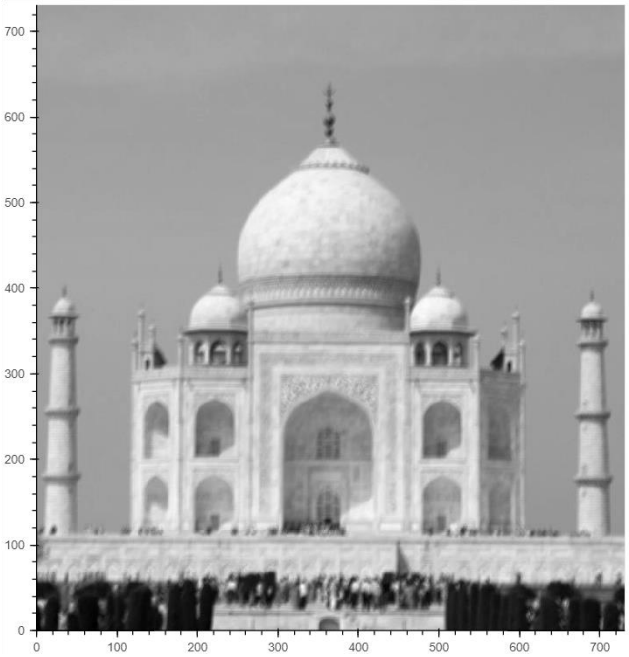



Cb




Cr

J'ai l'impression qu'on ne retrouve pas le flou de l'image sur ces deux canaux. Ces images ont effectivement d'autre problèmes de qualité mais si on déconvolu ces images on ne trouvera pas le bon noyau qui semble être très différent de celui de l'image Y en noir et blanc :



Y

Ainsi on obtiendrai un mauvais résultat.