INTRODUCTION AUX PROBLEMES INVERSES EN TRAITEMENT D'IMAGE

RESTAURATION D'IMAGES

Guénon Marie et Favreau Jean-Dominique

VIM / MASTER SSTIM

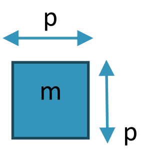
Table des matières

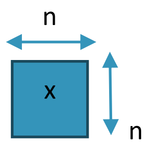
Question 1

En règle général, on cherche à utiliser un unique filtre m et un unique bruit de réalisation b. Mais si le système ne répond pas de la même manière à toutes les longueurs d'ondes alors on peut utiliser plusieurs filtres pour avoir un résultat plus précis.

Question 2

Soit une image x et un filtre m représentés sous forme matricielle telle que :





Avec p<<n. On cherche à faire leur produit de convolution en passant par la transformée de Fourier(FT)[1] :

$$m * x = FT^{-1}(FT(m).FT(x))$$

Ce qui pose un problème puisque les deux matrices ne sont pas de la même taille. Nous allons donc combler la différence de taille n – p entre l'image x et m en rajoutant dans l'image m des 0 :

$$\widetilde{m} = \begin{pmatrix} m & 0 \\ 0 & 0 \end{pmatrix}$$

On applique alors la FT sur x ainsi que sur $\widetilde{m}$ qui ont dorénavant la même taille. Une fois ceci fait, il suffit de faire la multiplication membre a membre des deux matrices obtenues puis la transformée de Fourier inverse pour obtenir le résultat que l'on recherche.

[1] On rappelle que la transformée de Fourier s'écrit : (X la transformée de Fourier de x)

$$X(k) = \sum_{j=0}^{n-1} x(j) e^{-2i\pi k \frac{j}{n}}$$

Question 3

Cet ajustement doit être effectué avant d'appliquer la transformée de Fourier discrète, car on ne change pas le filtre. Si on l'appliquait après la transformée de Fourier, on rajouterait un filtre passe-bas : lors du produit de X et M, toutes les hautes fréquences seraient supprimées.

Question 4

Deux options possibles pour récupérer les valeurs réelles d'une image seraient :

- La norme :

$$m * x = |FT^{-1}(FT(m).FT(x))|$$

- La partie Réelle :

$$m * x = Real\{FT^{-1}(FT(m).FT(x))\}$$

Dans notre cas, la méthode de la partie réelle est la plus adaptée car avec la norme nous perdrions l'information de signe, ce qui n'est pas souhaitable ici.

Question 5

Cette accumulation de convolution décale les pixels de l'image (shift) vers le bas à droite et rend l'image de plus en plus floue.

Pour k=1 ➔ on a un shift de $7 = \frac{p-1}{2}$

Pour k=10 ➔ on a un shift de 70

Le problème est dû à une translation de l'origine qui a eu lieu pendant la convolution.

Par ailleurs, on peut remarquer la "réapparition" des morceaux de l'image qui sont sortis du cadre à l'autre bout de l'image. Ceci est dû à la périodicité de la transformée de Fourier.

Question 6

Pour résoudre ce problème, nous avons remis le zéro de l'image et le zéro du filtre au même point central, et par périodicité, *m* est divisé en quatre sous parties qui sont affichées aux quatre coins de l'image.

m1	m2	0	0
m3	m4	0	0
0	0	0	0

Filtre initiale

m4	0	0	m3
0	0	0	0
m2	0	0	m1

Filtre Obtenu

Question 7

On cherche à "trouver *x* étant donné *m* et *y*" comme la minimisation de l'énergie suivante :

$$E = |y - m * x|^2 + \lambda |d * x|^2 (5)$$

Où $\lambda \geq 0$ et d est le filtre. (5) a pour solution unique :

$$X = \frac{M^*}{|M|^2 + \lambda |D|^2} Y$$

Où X et Y sont les transformée de Fourier respectives de x et y. Et M^* est le conjugué de M.

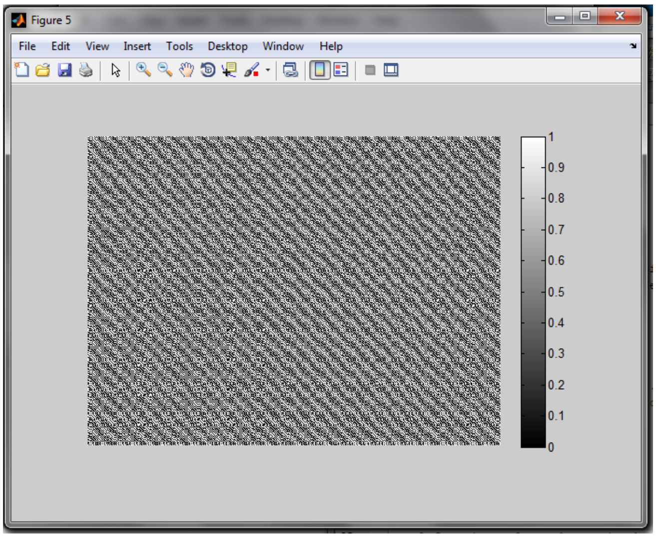
Nous utilisons ici $d = \delta$ (la fonction dirac) car $||d * x||^2 \xrightarrow{d=\delta} ||x||^2$ d'où $D = 1$

Nous avons alors :

$$X = \frac{M^*}{|M|^2 + \lambda} Y \ (6)$$

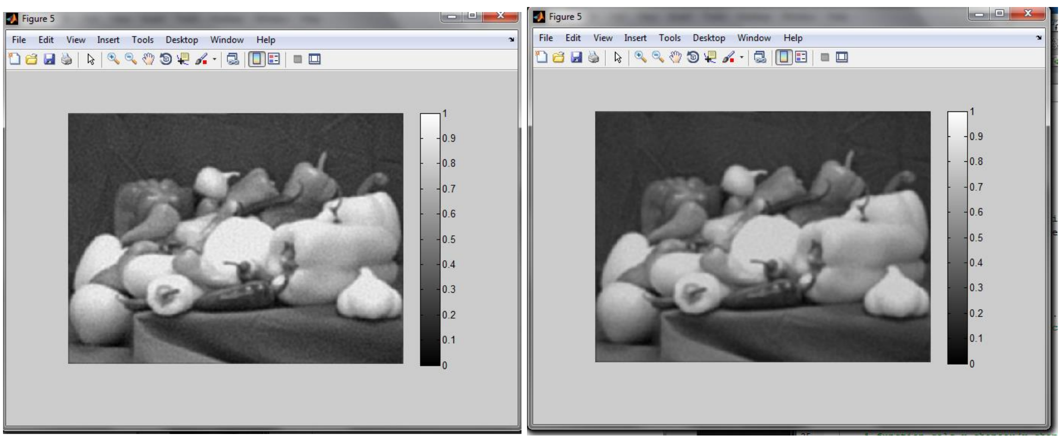
Question 8

Si on prend λ=0 alors on obtient le résultat suivant :

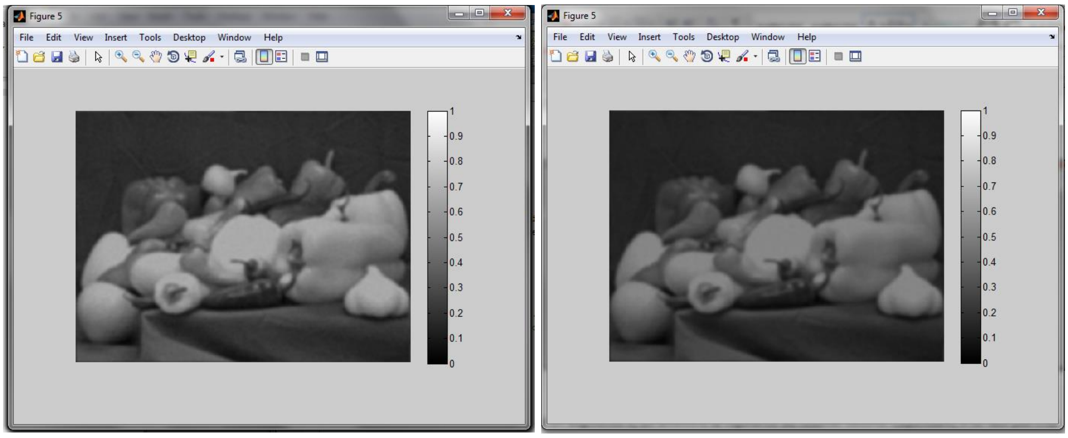


Ceci est dû au fait (5) se transforme en $E = |y - m * x|^2$ et nous cherchons le x qui minimise cette formule. En l'absence du terme de régularisation, il est normal que cohérence entre deux pixels côte à côte dans l'image ne soit pas conservée.
En effet, on remarque un effet similaire dans les cas suivants :

Question 9



Résultats obtenus pour λ=0.1 (à gauche) et λ=0.25 (à droite)



Résultats obtenus pour λ=0.5 (à gauche) et λ=0.9 (à droite)

Quand on augmente λ, on améliore la régularité du résultat obtenu. Cependant, il ne faut pas trop l'augmenter sinon l'imager se noircit au détriment de la visibilité.

Dans notre cas, le mieux est donc de prendre $0.1 \leq \lambda \leq 0.5$ si on n'applique pas d'autre traitement sur l'image.

Question 10

On prend maintenant :

$$E = |y - m * x|^2 + \lambda \int \varphi(|\nabla x|) \quad (7)$$

Où φ est une fonction qui préserve les propriétés de bord. Si m est radialement symétrique, on peut montrer que :

$$\nabla E(x) = 2\left[m * \big((m * x) - y\big) - \lambda\, div\left(\frac{\varphi'(|\nabla x|)}{2|\nabla x|}\nabla x\right)\right] \quad (8)$$

Minimiser (7) peut être fait en utilisant la méthode continue de descente de gradient suivant :

$$\begin{cases} x(\tau = 0) = x_0 \\ \frac{\partial_x}{\partial_\tau} = -\nabla E(x) \end{cases} \quad (9)$$

Une simple discrétisation de (9) mène à l'implémentation suivante :

$$\begin{cases} choisir\ x_0 \\ x_{n+1} = x_n - \alpha \nabla E(x_n) \end{cases}$$

Pour initialiser cette implémentation, nous pouvons ici choisir :

- $x_0 = y$
  Car y ne devrait pas être très loin de la solution que l'on cherche
- *Closed-form solution* (de 6) avec un bon λ
- $x_0 = zeros$ ou $x_0 = ones$ ou $x_0 = noise$,…

Nous pouvons choisir n'importe quelle initialisation x_0 puisque E (7) est convexe et possède un unique minimum. Cependant, il est préférable de prendre une initialisation proche du résultat que l'on cherche à atteindre pour avoir une convergnece rapide, et donc initialiser $x_0 = y$.

Question 11

Dans (8) On pose $c(x) = \frac{\varphi'(|\nabla x|)}{2|\nabla x|}$

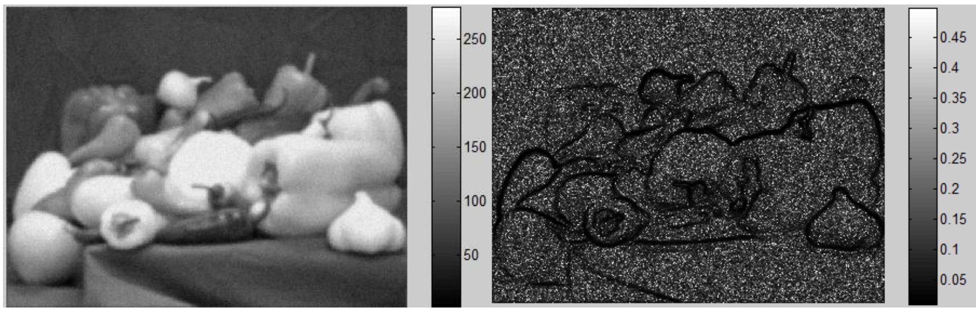
c est une représentation de bords (contour). A chaque point de l'image, c associe une valeur : plus un contour est fort (marqué) plus la valeur de c sera faible. Et inversement.

Question 12

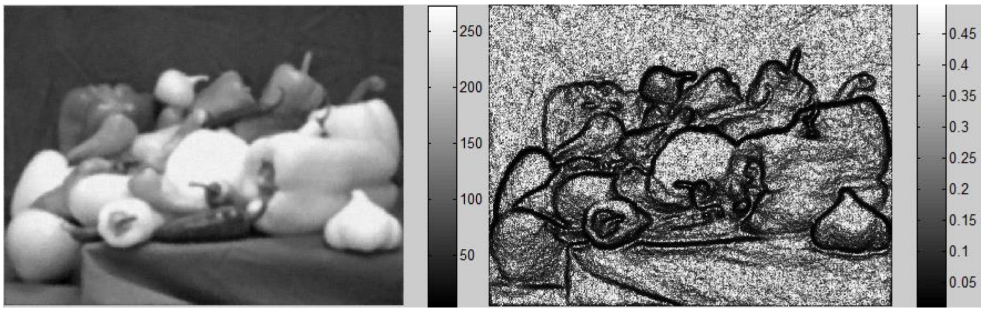
Pour calculer la variation de l'évolution, on peut calculer la norme de s(t+1)-s(t) qui est aussi la norme de $\alpha\nabla E$.

Question 13

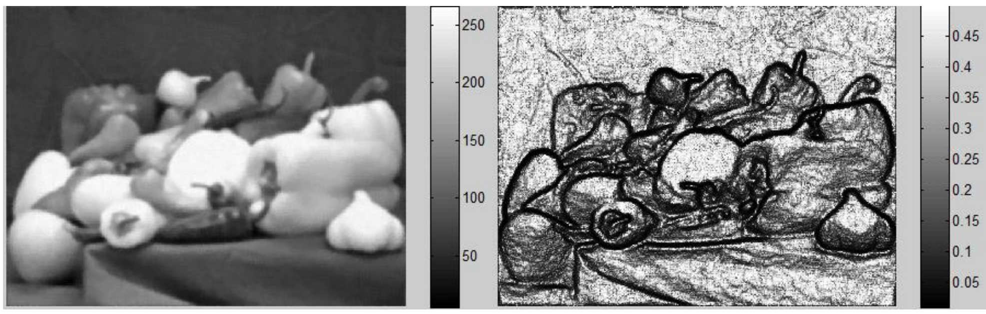
Ci-après les résultats obtenus après un certain nombre d'itérations. A gauche $c(x)$ et à droite s.



10 itérations



50 itérations

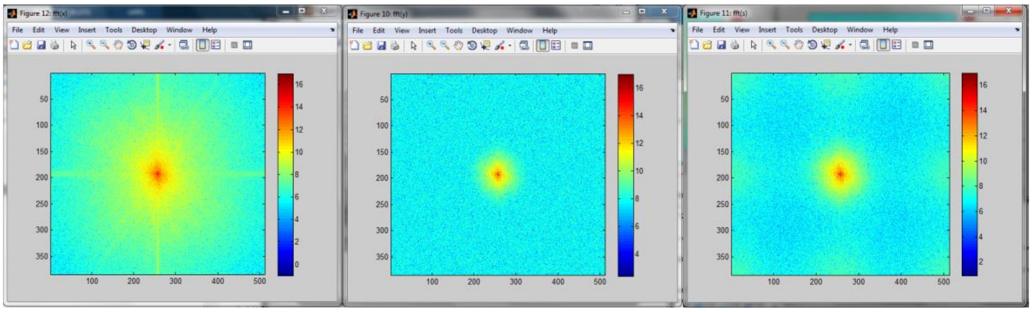


70 itérations

Comme nous pouvons le voir sur les images ci-dessus, au fil des itérations, nous obtenons un résultat de plus en plus précis et nous distinguons de mieux en mieux les contours.

Question 14

Transformées de Fourier obtenues avec $\alpha = 0{,}01$; à gauche celle de x, au milieu celle de y et à droite celle de s :



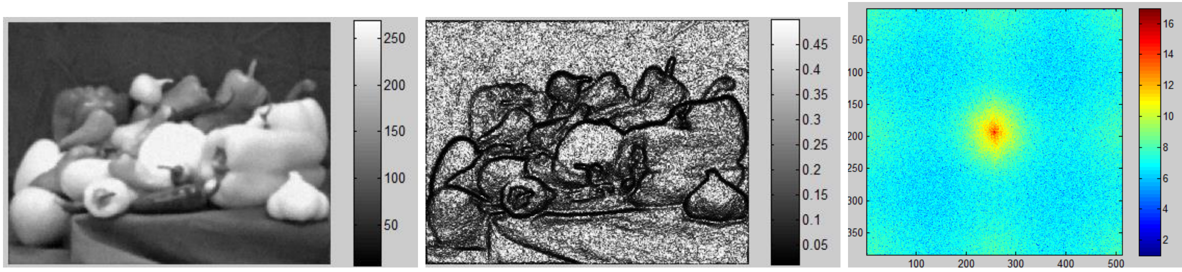
Dans *x* (image d'origine) on a des contours plus marqués, ce qui se voit dans la FFT qui est plus jaune.

Alors que dans *y* il y a des points partout car il y a du bruit (que nous avions rajouté q° 1) et donc rajoute de la discontinuité et des contours un peu partout

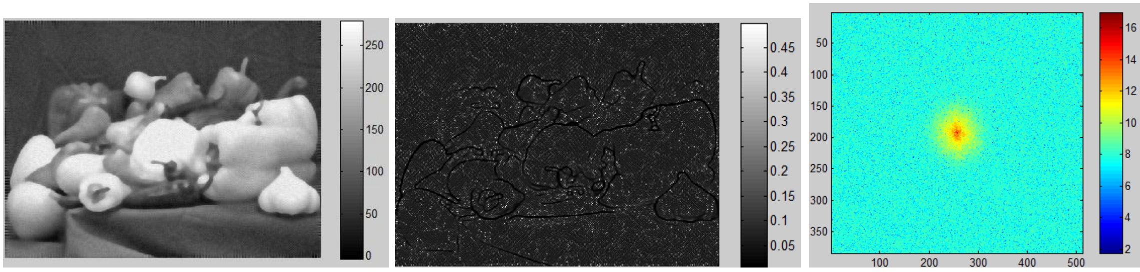
Et dans *s* il y a des points partout aussi mais il y a surtout plus de bleu sur le cercle médian car nous avons augmenté la smoothness de l'image. De plus on remarque sur la périphérie (sur les hautes fréquences) des taches jaunes qui sont restées de *y*, et qui n'ont pas disparu au traitement.

Question 15

Nous appliquons notre algorithme avec 50 itérations et obtenons les résultats suivants, s à gauche, c au milieu et la transformée de Fourier de s à droite :



$\boldsymbol{\alpha = 0,01}$



$\boldsymbol{\alpha = 0,5}$

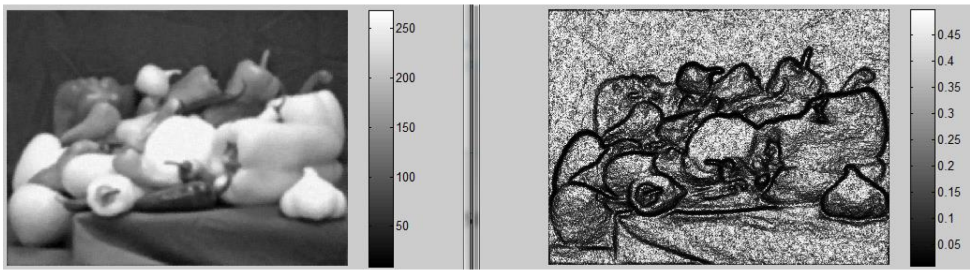
$\boldsymbol{\alpha = 1}$

$\boldsymbol{\alpha = 1,2}$

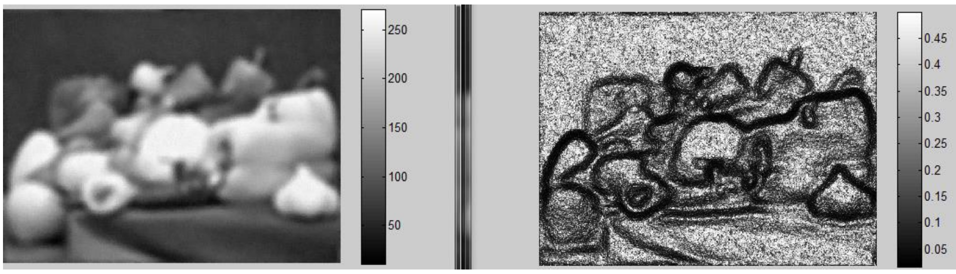
Nous pouvons remarquer que si nous augmentons trop α, on obtient une image reconstituée très floutée. De plus, nous avons remarqué que l'image se dégrade très rapidement dès que l'on prend un $\alpha > 1$, il est donc conseiller de se contenter de prendre un $\alpha \leq 1$…

Question 16

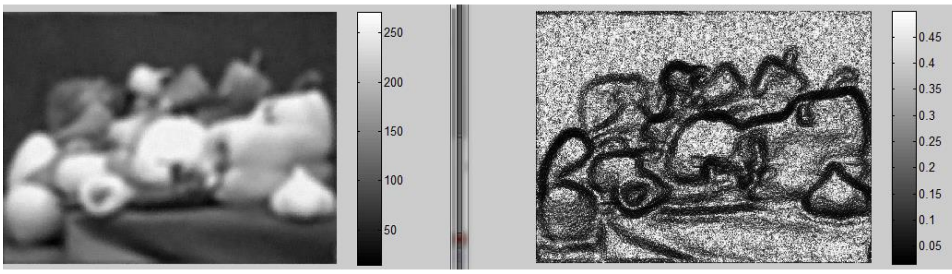
Dans ce cas, nous prenons $\alpha = 0{,}01$ et 50 itérations et nous faisons varier l'écart type (standard deviation, std). Nous obtenons les résultats suivant avec S à gauche et C à droite :



Std=2



Std=10



Std=50

Nous pouvons voir que sur l'image *c* les contours sont beaucoup plus épais, ce qui donne sur l'image *s* un résultat beaucoup plus flouté.