



POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE

ELE8812 – Rapport de laboratoire 2

Traitements élémentaires dans le domaine fréquentiel

Julien ANTOINE
1813026

Maxime SCHMITT
1719088

10 février 2016

1 Introduction

Dans le laboratoire précédent, nous avons effectué des manipulations sur des images en réalisant des manipulation sur celle-ci dans le domaine spatial. Dans le présent laboratoire, l'objectif est maintenant de réaliser des modifications aux résultats comparables mais en travaillant cette fois dans le domaine fréquentiel.

La première partie est dédiée à une familiarisation avec la représentation fréquentielle d'une image. La deuxième partie vise à rehausser une image au travers de manipulations de son spectre. La troisième et dernière partie a pour objectif de retirer le tramage d'une image à l'aide d'un filtre de réjection que l'on doit adapter à l'image considérée.

2 Représentation fréquentielle

2.1 Indexation des axes fréquentiels

La figure 2 illustre la représentation fréquentielle de l'image *Voiture.tif* en coordonnées réduites, on propose également en figure 1 la représentation standard à titre de comparaison.

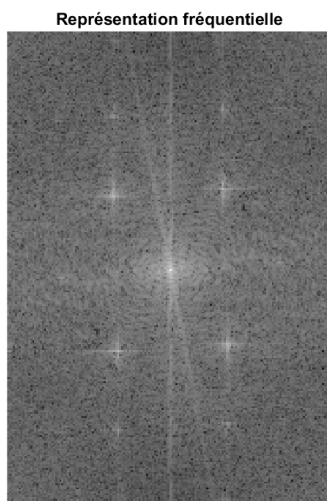


FIGURE 1 – Représentation fréquentielle de l'image sans transformation

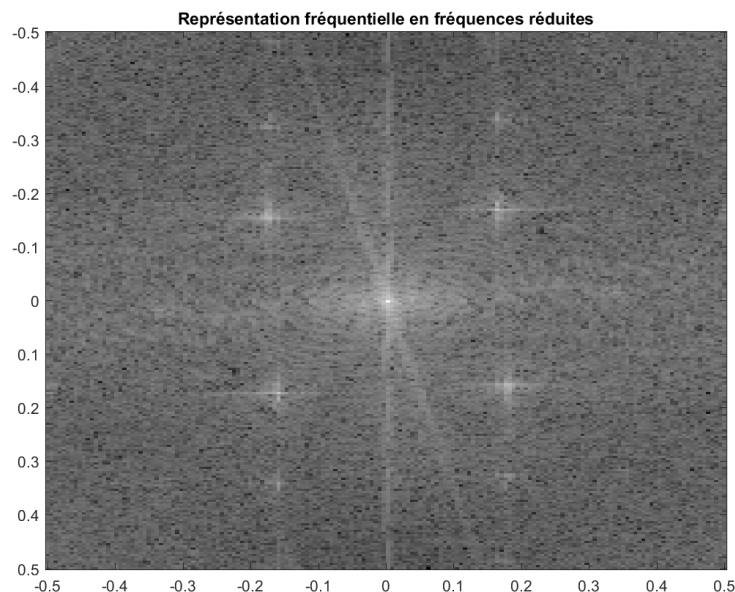


FIGURE 2 – Représentation fréquentielle en coordonnées réduites de l'image

2.2 Effet de la parité des dimensions de l'image

Les figures 3 et 4 représentent, dans le cas d'une image de dimension respectivement paire et impaire, la comparaison des résultats de la reconstruction de l'image *Voiture.tif* en utilisant *ifftshift* ou *fftshift*. On voit que l'utilisation de *fftshift* au lieu d'*ifftshift* est possible lorsque l'image est de dimension paire mais pas lorsqu'elle est de dimension impaire puisqu'on voit apparaître d'importants artefacts dans l'image reconstruite. Cela est dû entre autres au placement de la fréquence nulle qui n'est pas la même dans les deux cas lors du premier appel à *fftshift*. En effet, lorsque la dimension est paire, la fréquence nulle est centrée, les autres fréquences sont symétriques et il n'y a pas de fréquence en plus ou moins $1/2$. Au contraire, lorsque la dimension est impaire, la fréquence nulle est décalée d'un échantillon, et les autres fréquences sont symétriques sauf la fréquence à $-1/2$. Les fonctions *ifftshift* et *fftshift* ne fonctionnent donc pas de la même manière dans ce cas.

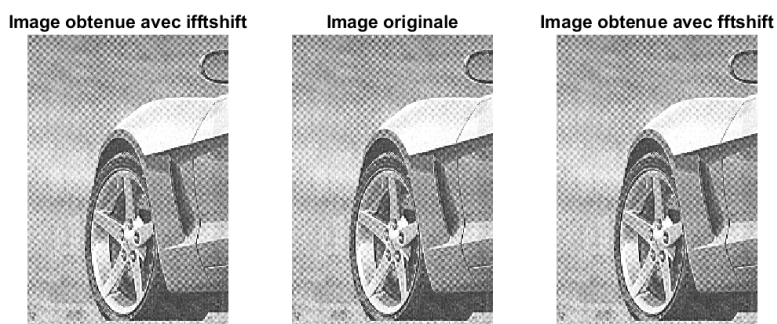


FIGURE 3 – Comparaison du résultat de l'utilisation de *ifftshift* ou *fftshift* sur la reconstruction de l'image dans le cas de dimensions paires

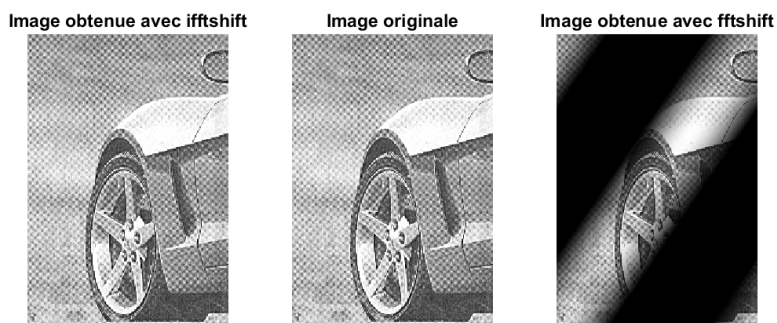


FIGURE 4 – Comparaison du résultat de l'utilisation de *ifftshift* ou *fftshift* sur la reconstruction de l'image dans le cas de dimensions impaires

3 Rehaussement dans le domaine de Fourier

3.1 Rehaussement par masquage flou

Une manière de rehausser une image est d'utiliser le masquage flou dans le domaine spectral. La méthode consiste à appliquer un filtre passe-bas à l'image, de manière à éliminer ses détails. La différence entre l'image originale et l'image filtrée donne une image des détails de l'image. Le principe est alors d'additionner cette image des différences à l'image originale afin de rehausser les détails et contours de celle-ci. Notre implémentation utilise un filtre de Butterworth d'ordre 2.

Rehaussement par masquage flou

```

1 % Paramètres
2 fc = 50; %
   Fréquence de coupure en pixels
3 type = 'gauss'; % Type du filtre (butter,ideal ou gauss
   )
4 pad = 0; % Nombre de pixels de padding
5 ordre = 2; % Ordre du filtre
6 c = 10; % Coefficient de masquage
7
8 %% Rehaussement par masquage flou
9 % Domaine de Fourier
10 [ims, HH] = filtpb(im, fc, type, pad, ordre); % Filtrage passe-
   bas
11 diff = im - ims; % Différence entre images originale
   et filtrée

```

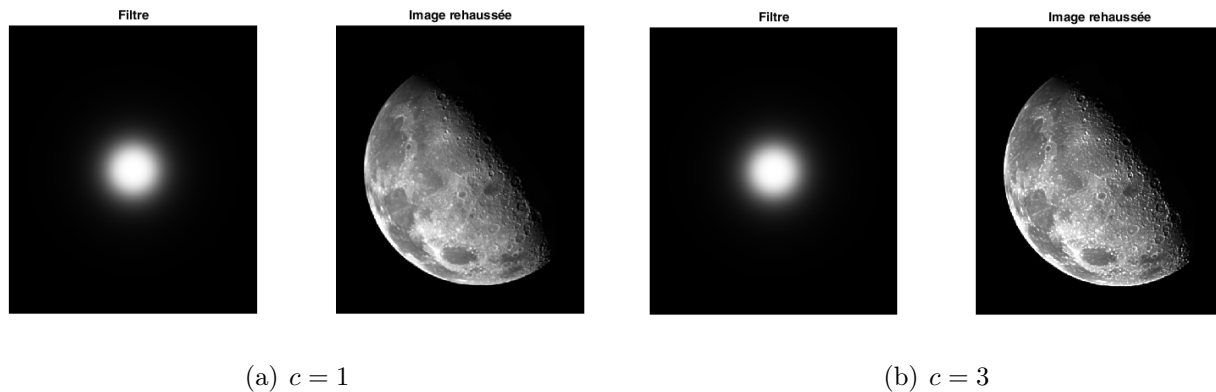
3.1.1 Variation de la fréquence de coupure

La figure 5 permet de remarquer que plus la fréquence de coupure augmente, et plus le rehaussement est faible. En effet, au plus la fréquence de coupure diminue, au plus de détails sont éliminés de l'image. En analysant la forme des filtres (à gauche), on constate que les fréquences coupées pour $f_c = 50$ sont bien moins importantes que celles pour $f_c = 5$. Dès lors, $f_c = 50$ rehaussera les détails dont les fréquences sont supérieures à 50 pixels, tandis que $f_c = 5$ rehaussera ceux avec des fréquences supérieures à 5 pixels, c'est-à-dire presque toute l'image y compris certaines textures.

FIGURE 5 – Influence de la fréquence de coupure à c constant ($c = 1$)

3.1.2 Variation du coefficient de masquage

Comme pour le premier travail pratique – les formules étant de même nature –, le coefficient de masquage définit l'intensité du rehaussement. Dès lors, pour une même fréquence de coupure, un coefficient de masquage c plus important rehaussera plus les détails de l'image, et inversement.

FIGURE 6 – Influence du coefficient de masquage c à fréquence de coupure constante ($f_c = 50$)

3.1.3 Comparaison entre approches spatiale et spectrale

Nous avons donc implémenté le rehaussement par masquage flou à la fois dans le domaine spatial et dans le domaine spectral. La figure 7 montre que les résultats obtenus sont très similaires, au point qu'il est impossible de distinguer de différence à l'œil nu. Elle a été obtenue avec les paramètres $\sigma = 2$ pour le domaine spatial, $f_c = 50$ pour le domaine spectral (un filtre gaussien a été utilisé cette fois afin de mieux correspondre à l'approche dans le domaine spatial), avec un coefficient de masquage $c = 3$ pour les deux.

Pour ce qui est de l'implémentation, celle dans le domaine spatial est plus simple à mettre en œuvre puisqu'il suffit d'appliquer un filtre gaussien à l'image et d'effectuer une sous-

traction pour obtenir le masque, là où dans le domaine spectral il faut d'abord calculer la transformée de Fourier, puis appliquer un filtre passe-bas avant d'effectuer la transformée de Fourier inverse et enfin la soustraction avec l'image originale. Si on rassemble ces opérations dans une fonction `filtpb`, la structure devient alors exactement la même.

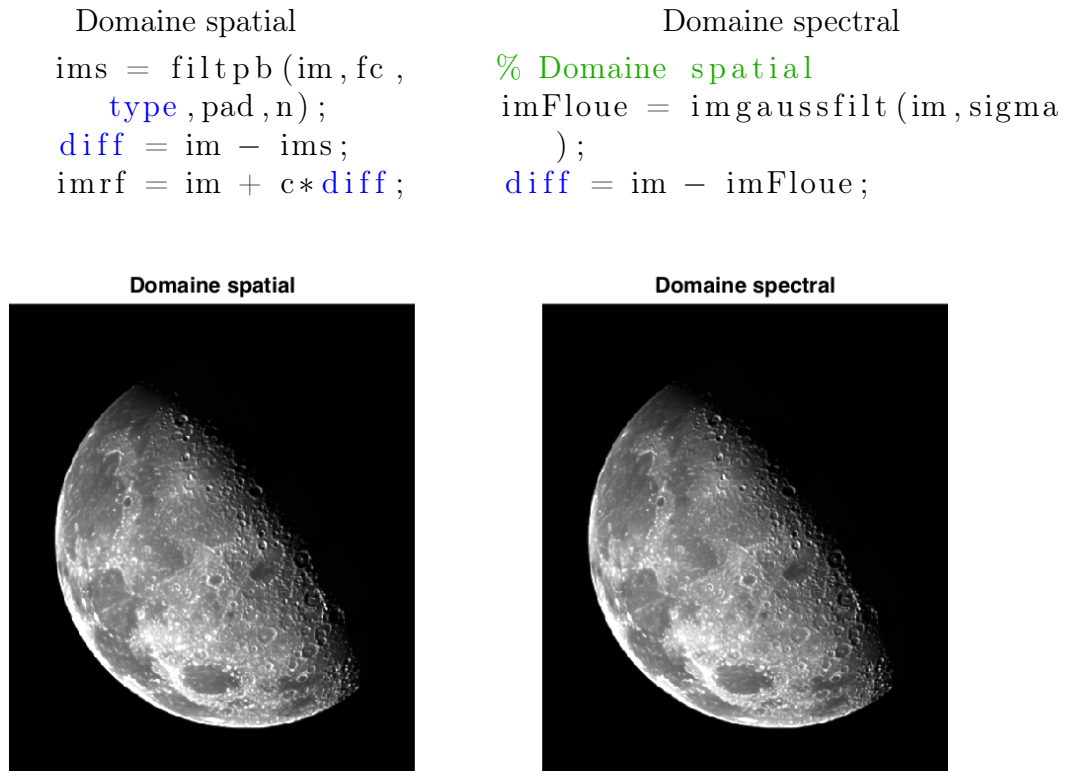


FIGURE 7 – Comparaison des résultats du rehaussement dans les 2 domaines

3.2 Rehaussement par laplacien

La conclusion est la même que pour le masquage flou : les résultats obtenus spatialement et fréquentiellement sont très similaires (voir figure 8), et l'implémentation dans le domaine spectral est plus ardue que dans le domaine spatial, de par la nature du laplacien en fréquentiel.

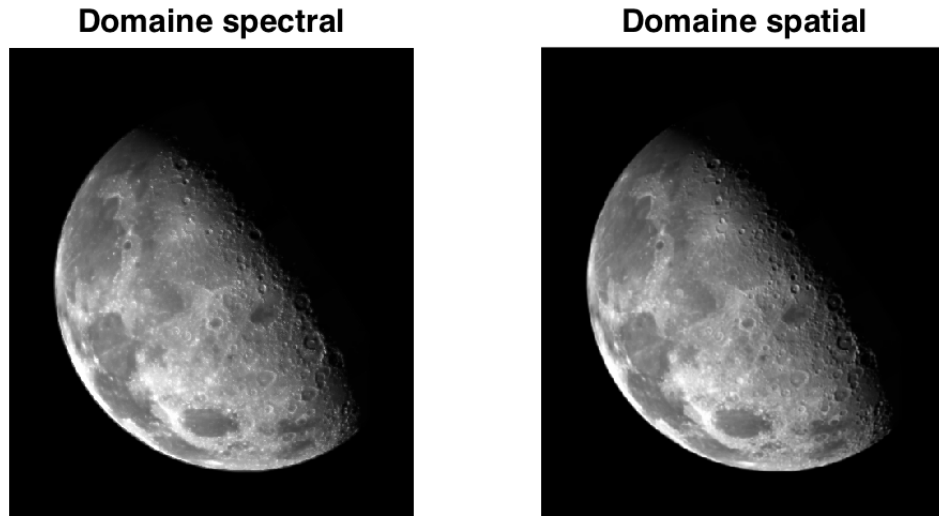


FIGURE 8 – Comparaison des résultats du rehaussement par laplacien dans les 2 domaines

```

L = [0 -1 0; -1 4 -1;
      0 -1 0];
laplacian = imfilter(
    im,L);
ims = im + c*laplacian
;

```

Domaine spatial

```

u0 = M/2; v0 = N/2;
[u,v] = ndgrid(1:M,1:N
);
duv = ((u-u0).^2 + (v-
v0).^2);
duv = duv/max(max(duv)
);
L = -4*pi^2*duv;
fuv = fftshift(fft2(im
));
fxy = ifft2(ifftshift(
L.*fuv));
imr = double(im) + c*
abs(fxy);
imr = uint8(imr);

```

Domaine spectral

3.2.1 Comparaison entre rehaussement par masquage flou et par laplacien

Tant bien pour les résultats que pour l'implémentation, l'approche par masquage flou nous paraît supérieure à celle par laplacien. Les résultats sont plus réalistes, et nécessitent moins d'attention aux échelles (normalisation). De plus, ils sont plus intuitifs, et dès lors plus facilement compréhensibles et modifiables. Enfin, on peut jouer sur plusieurs paramètres là où seul le coefficient de masquage permet de régler le rehaussement avec le laplacien.

4 Détramage par réjection de fréquences

Pour la construction du filtre de réjection, on a fait le choix pour la réponse fréquentielle du filtre de placer la fréquence nulle au voisinage du centre du tableau HH . Le code peut être trouvé en annexe C.

Pour effectuer le détramage de l'image *Voiture.tif*, on a tout d'abord construit le filtre coupant les fréquences représentées par les points les plus lumineux sur la représentation fréquentielle comme on peut l'observer sur la figure 9. Puis on a construit le filtre pour couper les fréquences représentées par les points moins lumineux sur la représentation fréquentielle comme on peut l'observer sur la figure 10. Le résultat de la combinaison de ces deux filtres peut être observé sur la figure 11, sur laquelle les trames dans l'images ont été lissées.

Pour définir les paramètres adéquats à l'utilisation de notre filtre de réjection sur l'image, on a calculé comme fréquence de coupure un "point moyen" des points à couper ainsi qu'une largeur de bande dont la valeur est fonction de la différence de distance maximale entre le centre et un des points à couper (le coefficient de proportionnalité pour cette relation a été trouvé empiriquement et dénote donc d'une préférence personnelle sur le résultat final).

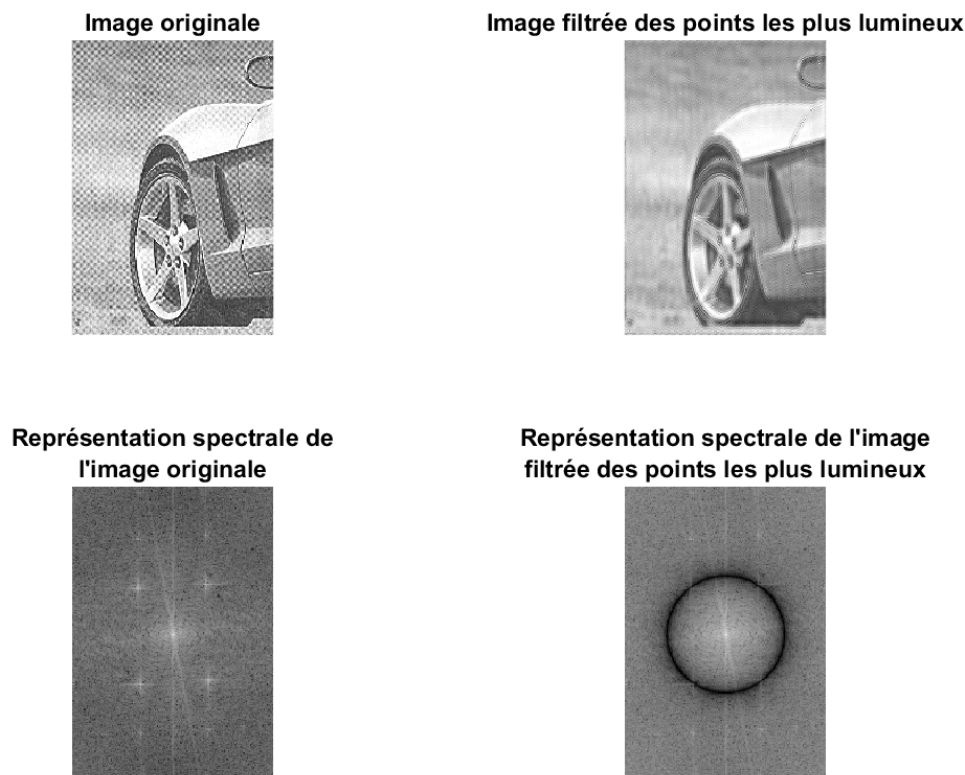


FIGURE 9 – Effet du filtrage des fréquences correspondant aux points les plus lumineux de la représentation fréquentielle

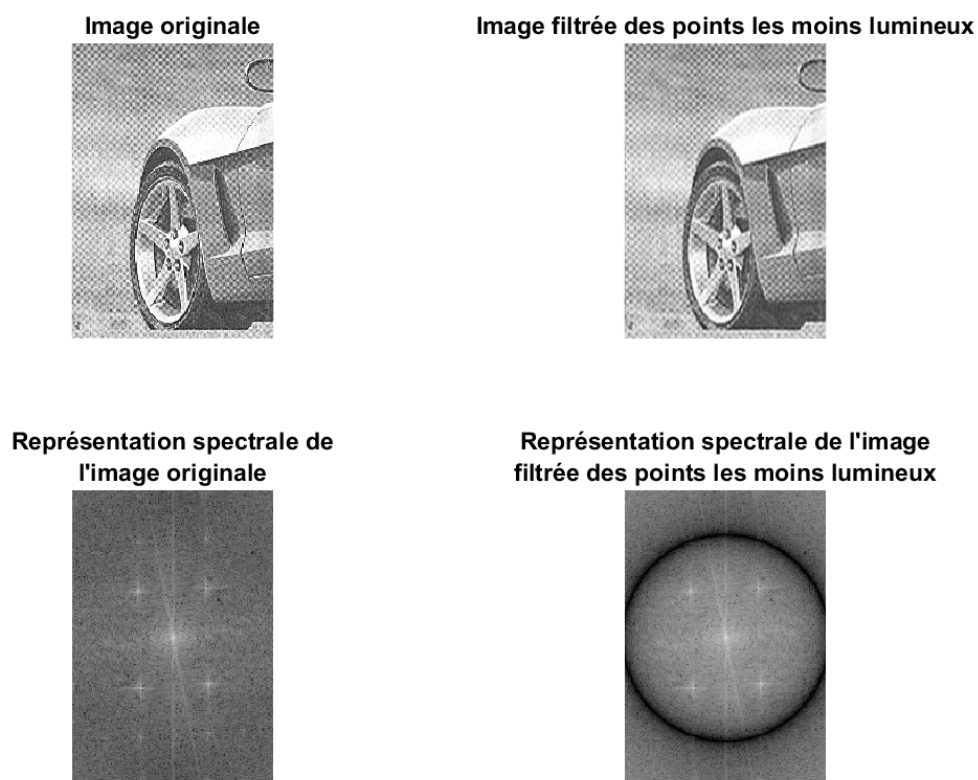


FIGURE 10 – Effet du filtrage des fréquences correspondant aux points les moins lumineux de la représentation fréquentielle

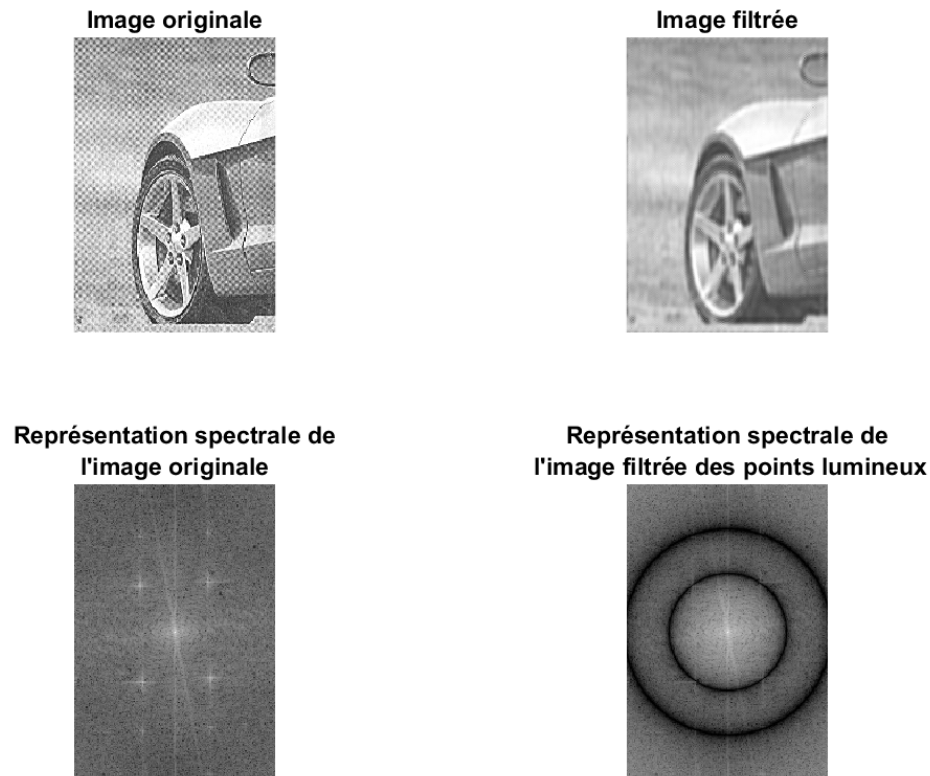


FIGURE 11 – Effet de la combinaison des filtres précédents

5 Conclusion

Au cours de ce laboratoire, nous avons eu l'occasion de nous familiariser avec la représentation fréquentielle d'une image et nous avons pu aborder quelques techniques de manipulation de celle-ci. Lorsqu'on en vient à comparer les résultats obtenus pour le rehaussement d'image par les manipulations sur le domaine spatial et ceux par les manipulations du domaine spectral, on constate que, sur les techniques abordées dans le cadre de ces laboratoires, ceux-ci sont équivalents et pourront donc être utilisés de façon interchangeable selon les spécificités de l'image. On retiendra aussi l'intérêt de la représentation spectrale pour obtenir de l'information supplémentaire sur l'image, que ce soit au niveau des motifs présents dans celle-ci mais également des artefacts qui pourraient y être présents, comme on a pu le voir dans le cas de la présence de trames.

Annexe A - Code partie 2 : representation.m

```

1  %
   *****%

2  % Fichier : representation.m
                                     %
3  % Description : Script effectuant la transformée de Fourier de l'
   image et %
4  %
   diverses manipulations sur celle-ci
                                     %
5  % Auteurs : Julien ANTOINE 1813026 & Maxime SCHMITT 1719088
                                     %
6  % Date : 10/02/2016
                                     %
7  %
   *****%

8
9  clear all; clc;
10
11 voiture = imread('Voiture.tif');
12 voitured = double(voiture);
13
14 %% 2.1 Indexation des axes fréquentiels
15
16 % Obtention de la représentation fréquentielle
17 F = fft2(voitured);
18
19 % Centrer la représentation fréquentielle
20 F = fftshift(F);
21 F2 = F; % On garde une version de la fft avec ses valeurs non
   transformées
22
23 % On rend la représentation affichable et lisible (utilisation du
   log pour
24 % utiliser une échelle de perception)
25 F = abs(F);
26 F = log(F+1);
27 F = mat2gray(F);
28
29 % Passage en fréquences réduites et affichage
30 figure(1);
31 imshow(F);
32 title('Représentation fréquentielle');

```

```

33 saveas(gcf, '../rapport/images/part2_freqBasiques.png');
34
35 figure(2);
36 colormap gray;
37 imagesc([-0.5 0.5],[-0.5 0.5],F);
38 title('Représentation fréquentielle en fréquences réduites');
39 saveas(gcf, '../rapport/images/part2_freqRéduites.png');
40
41 % Matrices des coordonnées des points lumineux
42 % PtsLumineux(:,1,[1 2]) -> coordonnées x et y en fréquences
    réduites
43 % PtsLumineux(:,2,[1 2]) -> coordonnées x et y en pixels
44 PtsLumineux = zeros(2,9,2);
45 % Coordonnées en fréquences réduites
46 PtsLumineux(1, 1, [1 2]) = [-0.1766 -0.3204];
47 PtsLumineux(2, 1, [1 2]) = [-0.1707 -0.1531];
48 PtsLumineux(3, 1, [1 2]) = [0.1647 -0.3367];
49 PtsLumineux(4, 1, [1 2]) = [0.1647 -0.1694];
50 PtsLumineux(5, 1, [1 2]) = [0.002994 0.002041];
51 PtsLumineux(6, 1, [1 2]) = [-0.1587 0.1735];
52 PtsLumineux(7, 1, [1 2]) = [-0.1587 0.3408];
53 PtsLumineux(8, 1, [1 2]) = [0.1766 0.1571];
54 PtsLumineux(9, 1, [1 2]) = [0.1826 0.3245];
55 % Coordonnées en pixels
56 PtsLumineux(1, 2, [1 2]) = [55 45];
57 PtsLumineux(2, 2, [1 2]) = [56 86];
58 PtsLumineux(3, 2, [1 2]) = [112 41];
59 PtsLumineux(4, 2, [1 2]) = [112 82];
60 PtsLumineux(5, 2, [1 2]) = [85 124];
61 PtsLumineux(6, 2, [1 2]) = [58 166];
62 PtsLumineux(7, 2, [1 2]) = [58 207];
63 PtsLumineux(8, 2, [1 2]) = [114 162];
64 PtsLumineux(9, 2, [1 2]) = [115 203];
65
66 % Affichage des données
67 for i=1:9
68     fprintf('Coordonnées du point brillant %d en fréquences
        réduites : (%.4f,%.4f) ; en pixels : (%d,%d)\n', i,
        PtsLumineux(i, 1, 1), PtsLumineux(i, 1, 2), PtsLumineux(i,
        2, 1), PtsLumineux(i, 2, 2));
69 end
70
71 %% 2.2 Effet de la parité des dimensions de l'image
72
73 % Comparaison de l'utilisation de fftshift ou ifftshift

```

```
74 figure(3);
75 subplot(1,3,1);
76 imshow(uint8(ifft2(ifftshift(F2))));
77 title('Image obtenue avec ifftshift');
78 subplot(1,3,2);
79 imshow(voiture);
80 title('Image originale');
81 subplot(1,3,3);
82 imshow(uint8(ifft2(fftshift(F2))));
83 title('Image obtenue avec fftshift');
84 saveas(gcf, '../rapport/images/part2_dimPairesComparaison.png');
85
86 % Même comparaison en enlevant un ligne et une colonne de l'image
    voiture
87 voitureTronquee = voiture(1:end-1,1:end-1);
88 voitureTronquee = double(voitureTronquee);
89 FTronquee = fft2(voitureTronquee);
90 FTronquee = fftshift(FTronquee);
91
92 figure(4);
93 subplot(1,3,1);
94 imshow(uint8(ifft2(ifftshift(FTronquee))));
95 title('Image obtenue avec ifftshift');
96 subplot(1,3,2);
97 imshow(voitureTronquee);
98 title('Image originale');
99 subplot(1,3,3);
100 imshow(uint8(ifft2(fftshift(FTronquee))));
101 title('Image obtenue avec fftshift');
102 saveas(gcf, '../rapport/images/part2_dimImpairesComparaison.png');
```

Annexe B - Code partie 3 : rehaussement.m

```

1  %
   *****%

2  % Fichier : rehaussement.m
                                     %

3  % Description : Script effectuant un rehaussement par masquage
   flou      %

4  %                                     puis par Laplacien (domaines
   spectral et spatial) %

5  % Auteurs : Julien ANTOINE 1813026 & Maxime SCHMITT 1719088
                                     %

6  % Date : 10/02/2016
                                     %

7  %
   *****%

8
9  % Acquisition de l'image
10 im = imread('Lune2.tif');
11 [M,N,~] = size(im);
12
13 % Paramètres
14 fc = 50;                                     %
   Fréquence de coupure en pixels
15 type = 'gauss';                             % Type du filtre (butter,ideal ou gauss
   )
16 pad = 0;                                     % Nombre de pixels de padding
17 ordre = 2;                                  % Ordre du filtre
18 c = 10;                                     % Coefficient de masquage
19
20 %% Rehaussement par masquage flou
21 % Domaine de Fourier
22 [ims, HH] = filtpb(im, fc, type, pad, ordre); % Filtrage passe-
   bas
23 diff = im - ims;                             % Différence entre images originale
   et filtrée
24 imrf = im + c*diff;                          % Rehaussement
25
26 % Paramètres (spatial)
27 sigma = 2;
28
29 % Domaine spatial
30 imFloue = imgaussfilt(im, sigma);

```

```

31 diff = im - imFloue;
32 imrs = im + c*diff;
33
34 % Affichage
35 subplot(1,2,1)
36 imshow(imrs)
37 title('Domaine spatial','FontSize',22)
38 subplot(1,2,2)
39 imshow(imrf)
40 title('Domaine spectral','FontSize',22)
41
42
43 %% Rehaussement par laplacien
44 % Domaine spectral
45 c = 0.25;
46 u0 = M/2; v0 = N/2; % Fréquences nulle
47 [u,v] = ndgrid(1:M,1:N);
48 duv = ((u-u0).^2 + (v-v0).^2); % Calcul de D(u,v) (p10/26 du
    chapitre 4)
49 duv = duv/max(max(duv)); % Normalisation de D(u,v)
50 laplacien = -4*pi^2*duv; % Calcul du Laplacien (p21
    /26 du chapitre 4)
51 fuv = fftshift(fft2(im)); % Calcul de la transformée
    de Fourier de l'image
52 fxy = ifft2(ifftshift(laplacien.*fuv)); % On repasse dans le
    domaine spatial
53 imr = double(im) + c*abs(fxy); % Rehaussement
54 % imr = imr/max(max(imr));
55 imr=uint8(imr);
56
57 % Domaine spatial
58 laplace = [0 -1 0; -1 4 -1; 0 -1 0]; % Noyau laplacien
59 % laplace = [-1 -1 -1; -1 8 -1; -1 -1 -1]; % Noyau laplacien
60 laplacian = imfilter(im,laplace); % Filtrage de l'
    image par le noyau
61 ims = im + c*laplacian; %
    Rehaussement
62
63
64 % Affichage
65 subplot(1,2,1)
66 imshow(imr)
67 title('Domaine spectral','FontSize',30)
68 axis off;
69 subplot(1,2,2)

```



```
70 imshow(ims)
71 title('Domaine spatial','FontSize',30)
72 axis off;
73
74 %% Enregistrement des fichiers
75 imwrite(imr,['SpectralC',num2str(c),'.png']);
76 imwrite(ims,['SpatialC',num2str(c),'.png']);
```

Annexe C - Code partie 4

rf_rej.m

```

1 function HH = rf_rej(nu1, nu2, W, M, N)
2 %
3 % Fonction calculant la réponse fréquentielle d'un filtre RÉEL 2D
   gaussien
4 % coupant les fréquences situées autour de (nu1, nu2) avec une "
   largeur
5 % de bande" W.
6 %
7 % nu1, nu2 et W sont exprimés en fréquences réduites (quantités
   comprises
8 % entre -0,5 et 0,5).
9 %
10 % Paramètres d'entrées
11 % nu1 : première coordonnée de la fréquence à rejeter
12 % nu2 : deuxième coordonnée de la fréquence à rejeter
13 % W : largeur de la bande à rejeter
14 % M : nombre de lignes de la réponse fréquentielle à synthétiser
15 % N : nombre de colonnes de la réponse fréquentielle à synthétiser
16 %
17 % Paramètre de sortie
18 % HH : réponse fréquentielle du filtre. Tableau 2D de taille (M, N
   )
19
20 % Construction du filtre passe-bas
21
22 % Coordonnées du centre de la représentation fréquentielle
23 M0 = ceil((M+1) / 2);
24 N0 = ceil((N+1) / 2);
25
26 [U, V] = ndgrid(1:M, 1:N);
27
28 % On fait le choix d'avoir HH tel que la fréquence nulle
29 % soit au voisinage de son centre
30 D2 = (U - M0).^2 + (V - N0).^2;
31
32 fc1 = sqrt((nu1*N)^2 + (nu2*M)^2);
33
34 % On construit et renvoie le filtre gaussien
35 HH = 1 - exp(-(D2 - fc1^2).^2 ./ ((W*(M+N)/2)^2 * D2));
36
37 end

```

detramage.m

```

1  %
   *****%
2  % Fichier : detramage.m
                                     %
3  % Description : Script effectuant le détramage pqr réjection de
   fréquences%
4  %                               de l'image Voiture.tif
                                     %
5  % Auteurs : Julien ANTOINE 1813026 & Maxime SCHMITT 1719088
                                     %
6  % Date : 16/02/2016
                                     %
7  %
   *****%

8
9  clear all; clc;
10
11 voiture = imread('Voiture.tif');
12 voitured = double(voiture);
13
14 % Obtention de la représentation fréquentielle
15 F = fftshift(fft2(voitured));
16
17 %% Entrée des coordonnées des fréquences à couper (cf question 2)
18
19 % Matrices des coordonnées des points lumineux
20 % PtsLumineux(:,1,[1 2]) -> coordonnées x et y en fréquences
   réduites
21 % PtsLumineux(:,2,[1 2]) -> coordonnées x et y en pixels
22 PtsLumineux = zeros(2,9,2);
23 % Coordonnées en fréquences réduites
24 PtsLumineux(1, 1, [1 2]) = [-0.1766 -0.3204];
25 PtsLumineux(2, 1, [1 2]) = [-0.1707 -0.1531];
26 PtsLumineux(3, 1, [1 2]) = [0.1647 -0.3367];
27 PtsLumineux(4, 1, [1 2]) = [0.1647 -0.1694];
28 PtsLumineux(5, 1, [1 2]) = [0.002994 0.002041];
29 PtsLumineux(6, 1, [1 2]) = [-0.1587 0.1735];
30 PtsLumineux(7, 1, [1 2]) = [-0.1587 0.3408];
31 PtsLumineux(8, 1, [1 2]) = [0.1766 0.1571];
32 PtsLumineux(9, 1, [1 2]) = [0.1826 0.3245];
33 % Coordonnées en pixels

```

```

34 PtsLumineux(1, 2, [1 2]) = [55 45];
35 PtsLumineux(2, 2, [1 2]) = [56 86];
36 PtsLumineux(3, 2, [1 2]) = [112 41];
37 PtsLumineux(4, 2, [1 2]) = [112 82];
38 PtsLumineux(5, 2, [1 2]) = [85 124];
39 PtsLumineux(6, 2, [1 2]) = [58 166];
40 PtsLumineux(7, 2, [1 2]) = [58 207];
41 PtsLumineux(8, 2, [1 2]) = [114 162];
42 PtsLumineux(9, 2, [1 2]) = [115 203];
43
44 %% Coupure des fréquences les plus significatives (centre de la
    représentation fréquentielle)
45
46 % On ne le fait pas ici (l'énoncé demande de filtrer les points
    lumineux
47 % mais celui-ci n'est pas un artefact du type que l'on veut
    supprimer)
48
49 % [~, HH1] = filtpb(voitured, 5, 'gauss');
50 % HH1 = ones(size(voitured)) - HH1;
51 % imFilt1 = uint8(ifft2(ifftshift(F .* HH1)));
52 %
53 % figure(1);
54 % subplot(1,2,1);
55 % imshow(mat2gray(log(abs(F)+1)));
56 % subplot(1,2,2);
57 % imshow(mat2gray(log(abs(F.*HH1)+1)));
58 %
59 % figure(2);
60 % subplot(1,2,1);
61 % imshow(voiture);
62 % title('Image originale');
63 % subplot(1,2,2);
64 % imshow(imFilt1);
65 % title('Image filtrée');
66
67 %% Coupure des secondes fréquences les plus significatives ("
    Premier cercle" : points d'indice pair)
68
69 % On détermine W en trouvant la largeur de l'anneau contenant tous
    les
70 % points lumineux
71
72 % On calcule la distance de chaque point au centre
73 distances = zeros(4,1);

```

```

74     for i=[2 4 6 8]
75         distances(i/2) = sqrt(PtsLumineux(i,1,1)^2 + PtsLumineux(i
            ,1,2)^2);
76     end
77     maxDist = max(distances);
78     minDist = min(distances);
79
80     % On choisit comme point où appliquer le filtre un point "
        moyen" des points
81     % concernés
82     nu1 = mean(abs(PtsLumineux([2 4 6 8],1,1)));
83     nu2 = mean(abs(PtsLumineux([2 4 6 8],1,2)));
84
85     % On choisit comme largeur de bande en fonction de la
        différence de
86     % distance entre le point le plus éloigné et celui le plus
        proche
87     % (Chiffre 40 trouvé empiriquement; relève d'une préférence
        personnelle du résultat)
88     W = (maxDist - minDist) * 40;
89
90     % On applique le filtre
91     HH2 = rf_rej(nu1, nu2, W, size(voiture,1), size(voiture,2));
92     imFilt2 = uint8(ifft2(ifftshift(F .* HH2)));
93
94     % Affichage du résultat
95     figure(1);
96     subplot(2,2,1);
97     imshow(voiture);
98     title('Image originale');
99     subplot(2,2,2);
100    imshow(imFilt2);
101    title('Image filtrée des points les plus lumineux');
102    subplot(2,2,3);
103    imshow(mat2gray(log(abs(F)+1)));
104    title({'Représentation spectrale de ' 'l''image originale'});
105    subplot(2,2,4);
106    imshow(mat2gray(log(abs(F.*HH2)+1)));
107    title({'Représentation spectrale de l''image' 'filtrée des points
        les plus lumineux'});
108
109    % Sauvegarde de la figure
110    saveas(gcf, '../rapport/images/part4_filtreBas.png');
111
112    %% Coupure des dernières fréquences les plus significatives ("

```

```

    Deuxième cercle" : points d'indice impair sauf 5)
113
114 % On détermine W en trouvant la largeur de l'anneau contenant tous
    les
115 % points lumineux
116
117 % On calcule la distance de chaque point au centre
    distances = zeros(4,1);
118
119 for i=[1 3 7 9]
120     distances(logical(floor(i/2))) = sqrt(PtsLumineux(i,1,1)^2
        + PtsLumineux(i,1,2)^2);
121
122 end
123 maxDist = max(distances);
124 minDist = min(distances);
125
126 % On choisit comme point où appliquer le filtre un point "
    moyen" des points
127 % concernés
128 nu1 = mean(abs(PtsLumineux([1 3 7 9],1,1)));
129 nu2 = mean(abs(PtsLumineux([1 3 7 9],1,2)));
130
131 % On choisit comme largeur de bande en fonction de la
    différence de
132 % distance entre le point le plus éloigné et celui le plus
    proche
133 W = (maxDist - minDist);
134
135 % On applique le filtre
136 HH3 = rf_rej(nu1, nu2, W, size(voiture,1), size(voiture,2));
137 imFilt3 = uint8(fftshift(fft(F .* HH3)));
138
139 % Affichage du résultat
140 figure(2);
141 subplot(2,2,1);
142 imshow(voiture);
143 title('Image originale');
144 subplot(2,2,2);
145 imshow(imFilt3);
146 title('Image filtrée des points les moins lumineux');
147 subplot(2,2,3);
148 imshow(mat2gray(log(abs(F)+1)));
149 title({'Représentation spectrale de ' 'l''image originale'});
150 subplot(2,2,4);
151 imshow(mat2gray(log(abs(F.*HH3)+1)));
152 title({'Représentation spectrale de l''image', 'filtrée des points

```

```
        les moins lumineux'}));  
152  
153 % Sauvegarde de la figure  
154 saveas(gcf, '../rapport/images/part4_filtreHaut.png');  
155  
156 %% Image filtrée avec tous les filtres  
157  
158 figure(3);  
159 subplot(2,2,1);  
160 imshow(voiture);  
161 title('Image originale');  
162 subplot(2,2,2);  
163 imshow(uint8(ifft2(ifftshift(F .* HH2 .* HH3))));  
164 title('Image filtrée');  
165 subplot(2,2,3);  
166 imshow(mat2gray(log(abs(F)+1)));  
167 title({'Représentation spectrale de' 'l''image originale'});  
168 subplot(2,2,4);  
169 imshow(mat2gray(log(abs(F .* HH2 .* HH3)+1)));  
170 title({'Représentation spectrale de' 'l''image filtrée des points  
        lumineux'}));  
171  
172 % Sauvegarde de la figure  
173 saveas(gcf, '../rapport/images/part4_filtreCompleet.png');
```