

—> on remarque des artefacts aux bords...
=> solution : zero-padding !

- **Exercice 16 :** Reprendre l'exercice 15 avec un rayon du disque convoluant de 24px et en appliquant un pavage de zéro dans des tableaux deux fois plus grands (linéairement).

```

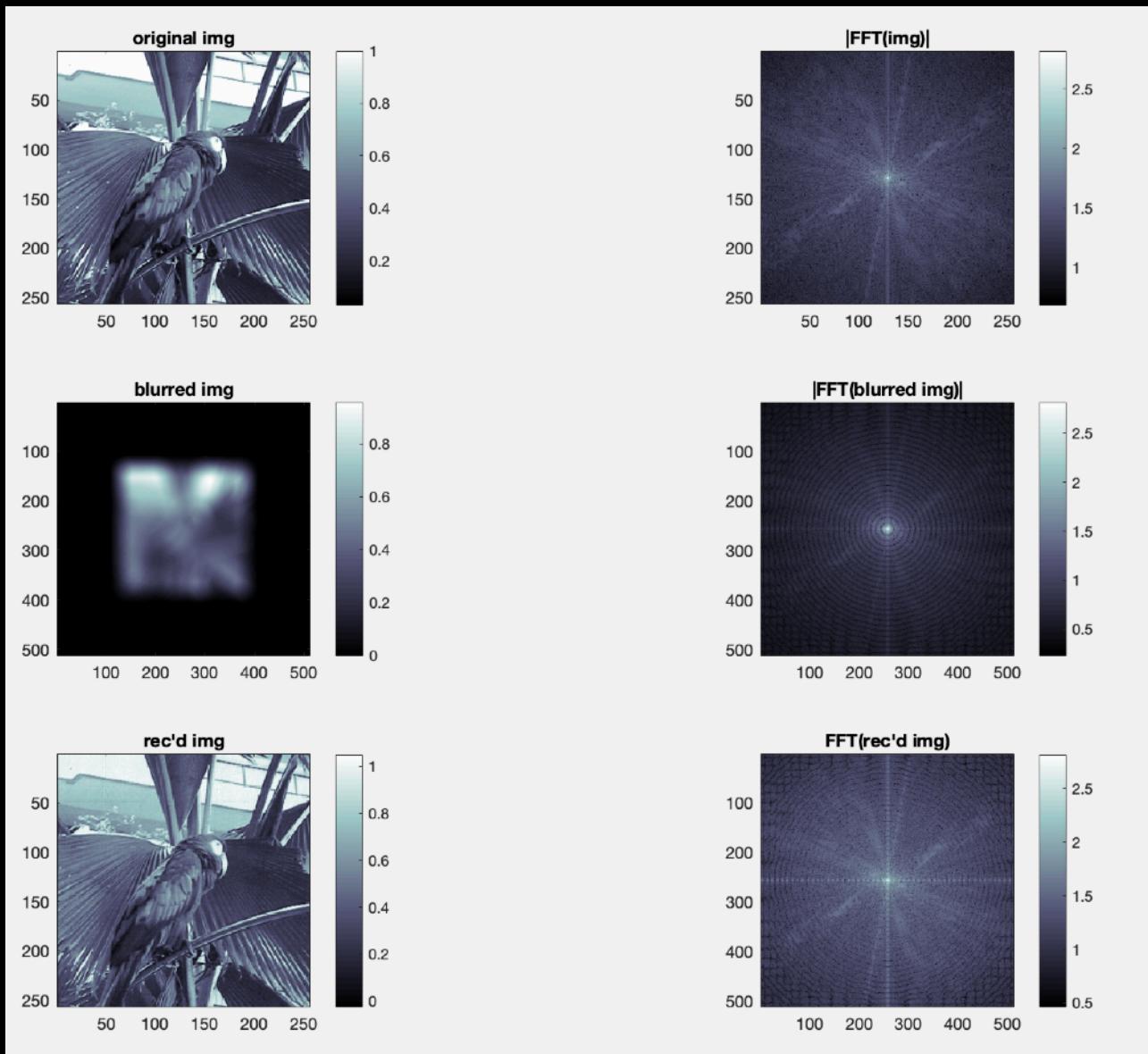
1 %%%%%%
2 %%% FILTRAGE INVERSE AVEC ZERO_PADDING %%%
3 %%%%%%
4
5 %%% préliminaires
6 clear
7 close all
8
9 %%%%%%
10 %%% PREMIÈRE PARTIE : SIMULATION %%%
11 %%%%%%
12
13 %%% préparation image
14 img=imread('/Users/marcel/Documents/MATLAB/0-images/bird.jpg');
15 img=rgb2gray(img); img=double(img)/255.;
16 dim=size(img); dimx=dim(1); dimy=dim(2);
17 img_chap = fft2(img,2*dimx,2*dimy); % FFT (+zero-padding au passage)
18
19 %%% floutage (convolution par un disque) (+zero-padding au passage)
20 rr=24; gag=fspecial('disk',rr);
21 gau=zeros(dim); gau(dimx/2+1-rr:dimx/2+1+rr,dimy/2+1-rr:dimy/2+1+rr)=gag;
22 dummy=conv2(img,gau,'full'); imgg=zeros(2*dimx,2*dimy); imgg(2:end,2:end)=dummy;
23
24 %%% solution alternative (dans le plan de Fourier)
25 gag_chap=fft2(gag,2*dimx,2*dimy);
26 imgg_chap=gag_chap.*img_chap;
27 imgg2 = ifft2(imgg_chap); % real() si besoin...

```

```

29 %%%%%%
30 %%% DEUXIÈME PARTIE : TRAITEMENT %%%
31 %%%%%%
32
33 %%% défloutage par filtrage inverse
34 gau_chap=fft2(gau,2*dimx,2*dimy);
35 imgg_chap=fft2(imgg);
36 seuil = .001; % on peut jouer ici avec le seuil pour un résultat optimal
37 idx=find(abs(gau_chap) < seuil);
38 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
39 img_rec_chap=imgg_chap./gau_chap;
40 img_rec=real(ifft2(img_rec_chap));
41 img_rec=img_rec(1:dimx,1:dimy); % pas besoin de réordonner les quadrants
42 % ici, on prend directement le bon
43 %%% affichage filtrage inverse avec zero padding
44 figure(), colormap('bone')
45 subplot(3,2,1), imagesc(img), colorbar
46 title('original img'), axis('image')
47 subplot(3,2,2), imagesc(abs(fftshift(fft2(img))).^1), colorbar
48 title('|FFT(img)|'), axis('image')
49 subplot(3,2,3), imagesc(imgg), colorbar
50 title('blurred img'), axis('image')
51 subplot(3,2,4), imagesc(abs(fftshift(imgg_chap)).^1), colorbar
52 title('|FFT(blurred img)|'), axis('image')
53 subplot(3,2,5), imagesc(img_rec), colorbar
54 title("rec'd img"), axis('image')
55 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
56 title("FFT(rec'd img)'), axis('square')

```



- Préférer à partir de maintenant la méthode de convolution en passant par Fourier pour le floutage.

- **Exercice 17 :** Reprendre l'exercice en prenant un disque convoluant de rayon appliquant à l'image floutée un bruit poivre et sel affectant seulement 2% des pixels. Optimiser le seuil lors de l'inversion afin de trouver un équilibre entre propagation du bruit et apparition d'artefacts. Puis tenter une réduction du bruit poivre et sel avant inversion et ré-optimiser le seuil. Comparer les deux seuils.

```

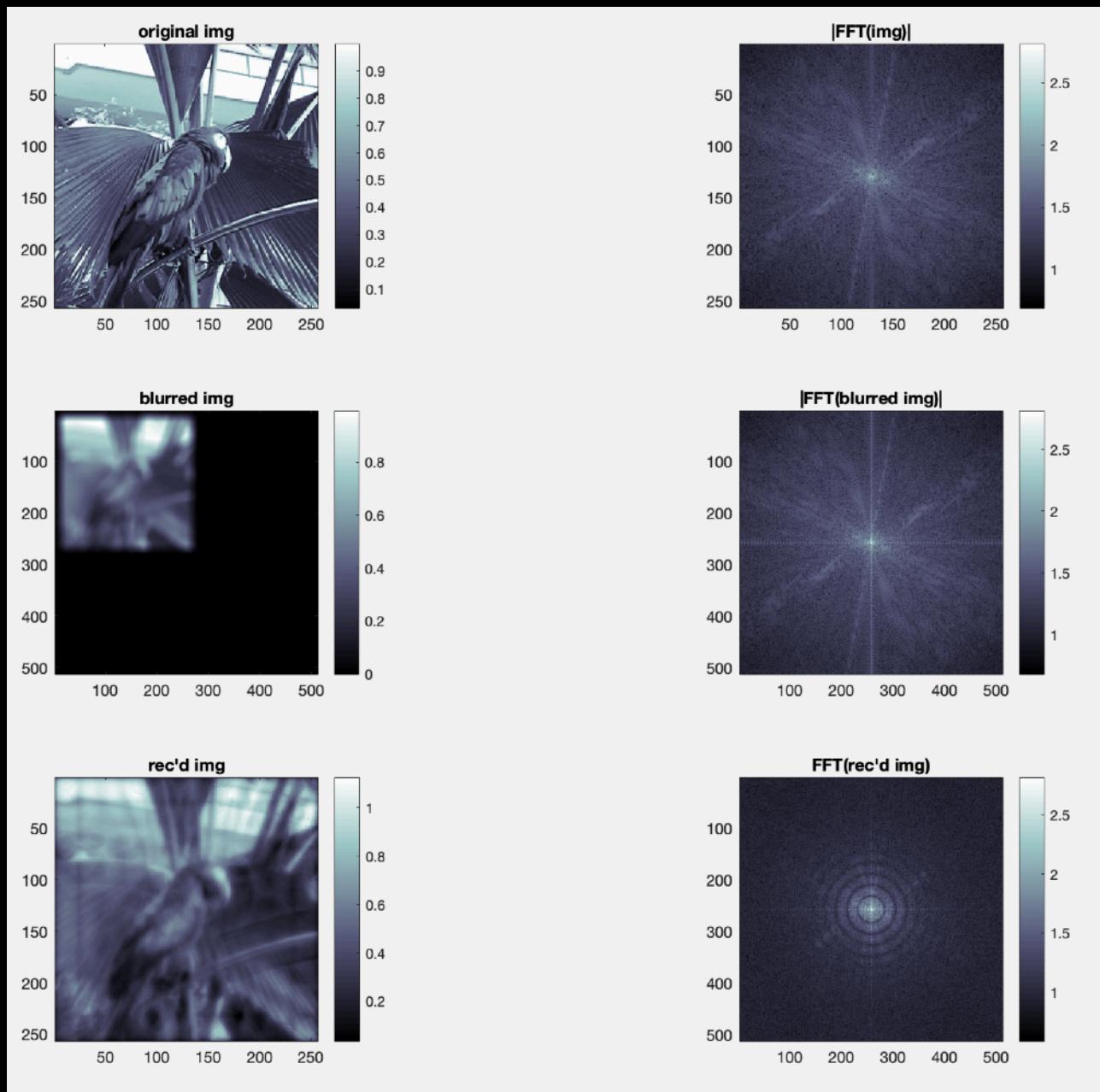
1  %% FILTERING INVERSE IN THE PRESENCE OF SALT & PEPPER NOISE
2
3
4
5  %% Preliminaries
6  clear
7  close all
8
9
10 %% FIRST PART: SIMULATION
11
12
13 %% Preparation of image
14 img = imread('/Users/marcel/Documents/MATLAB/0-images/bird.jpg');
15 img = rgb2gray(img); img = double(img)/255.;
16 dim = size(img); dimx = dim(1); dimy = dim(2);
17 img_chap = fft2(img, 2*dimx, 2*dimy);      % FFT(I) (with zero-padding)
18
19 %% Blurring (convolution with a disk) with zero-padding
20 %% (convolution passing through the Fourier plane)
21 rr = 12; gag = fspecial('disk', rr);
22 gag_chap = fft2(gag, 2*dimx, 2*dimy);      % FFT(G) (with zero-padding)
23 imgg_chap = gag_chap .* img_chap;           % blurring via Fourier
24 imgg = ifft2(imgg_chap);                    % I_foulo
25
26 %% Salt & Pepper noise at 2%
27 imgg = imnoise(imgg, 'salt & pepper', 0.02);

```

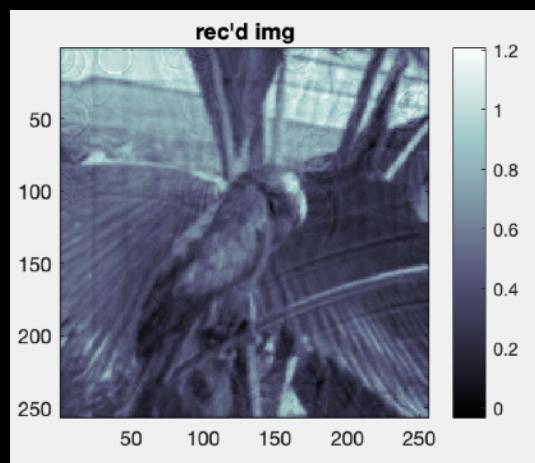
```

29 %% DEUXIÈME PARTIE : TRAITEMENT %%
30
31
32
33 %% filtre médian pour contrer le bruit p&s
34 imgg=medfilt2(imgg); % filtre médian sur I_floue bruitée
35
36 %% défloutage par filtrage inverse
37 gau_chap=gag_chap;
38 imgg_chap=fft2(imgg);
39 seuil = 0.2; % seuil à optimiser...
40 idx=find(abs(gau_chap) < seuil);
41 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
42 img_rec_chap=imgg_chap./gau_chap; % filtre inverse
43 img_rec=real(ifft2(img_rec_chap)); % real() nécessaire ici
44 img_rec=img_rec(1:dimx,1:dimy); % partie pertinente de img_rec
45
46 %% affichage filtrage inverse avec zero padding en présence de bruit p&s
47 figure(), colormap('bone')
48 subplot(3,2,1), imagesc(img), colorbar
49 title('original img'), axis('image')
50 subplot(3,2,2), imagesc(abs(fftshift(fft2(img))).^1), colorbar
51 title('|FFT(img)|'), axis('image')
52 subplot(3,2,3), imagesc(imgg), colorbar
53 title('blurred img'), axis('image')
54 subplot(3,2,4), imagesc(abs(fftshift(img_chap)).^1), colorbar
55 title('|FFT(blurred img)|'), axis('image')
56 subplot(3,2,5), imagesc(img_rec), colorbar
57 title("rec'd img"), axis('image')
58 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
59 title("FFT(rec'd img)"), axis('image')

```



avec $s=0.05 \rightarrow$



- **Exercice 18 :** Reprendre l'exercice en appliquant toujours un bruit poivre et sel (2% des pixels), mais avec de surcroît un bruit gaussien d'écart type relatif 0.05. Appliquer de nouveau le filtre non-linéaire permettant de réduire le bruit poivre et sel, et appliquer aussi un filtre pour contrer le bruit gaussien. Jouer toujours avec le seuil du filtre inverse pour trouver un équilibre entre artefacts et propagation du bruit.

```

1  %%%%%%
2  %%% FILTRAGE INVERSE AVEC P&S ET RON %%%
3  %%%%%%
4
5  %%% préliminaires
6  clear
7  close all
8
9  %%% préparation image
10 img = imread('/Users/marcel/Documents/MATLAB/0-images/bird.jpg');
11 img = rgb2gray(img); img = double(img)/255.; % image
12 dim = size(img); dimx = dim(1); dimy = dim(2); % dimensions de l'image
13 img_chap = fft2(img, 2*dimx, 2*dimy); % FFT(image) (+zero-padding)
14 fx = ((0:dimx-1)-dimx/2)/dimx; % étalonnage des fréquences
15 fy = ((0:dimy-1)-dimy/2)/dimy; % étalonnage des fréquences
16
17 %%% floutage (convolution par un disque) (+zero-padding au passage)
18 rr = 12; gag = fspecial('disk', rr); % disque
19 gag_chap = fft2(gag, 2*dimx, 2*dimy); % FFT(disque) (+zero padding)
20 imgg_chap = gag_chap .* img_chap; % floutage dans le plan de Fourier
21 imgg = ifft2(imgg_chap); % real() si nécessaire
22
23 %%% bruitage poivre et sel à 2%
24 imgg = imnoise(imgg, 'salt & pepper', 0.02); % bruit poivre et sel
25
26 %%% bruitage gaussien additif (sigma=0.05)
27 imgg = imnoise(imgg, 'gaussian', 0., 0.05^2); % bruit de lecture
28

```

```

30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 %% DEUXIÈME PARTIE : TRAITEMENT %%
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33
34 %% filtre médian pour contrer le bruit p&s
35 imgg_mf=medfilt2(imgg);
36
37 %% réduction du bruit gaussien par un filtre passe-bas gaussien
38 nn=25; gag=fspecial('gaussian',2*dim, nn);
39 imgg_mf_chap=fftshift(gag).*fft2(imgg_mf);
40
41 %% défloutage par filtrage inverse
42 gau_chap=gag_chap;
43
44 seuil = .1; % 1er seuil
45 idx=find(abs(gau_chap) < seuil);
46 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
47 img_rec_chap1=imgg_mf_chap./gau_chap;
48 img_rec=real(ifft2(img_rec_chap1)); img_rec1=img_rec(1:dimx,1:dimy);
49
50 seuil = .5; % 2e seuil
51 idx=find(abs(gau_chap) < seuil);
52 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
53 img_rec_chap2=imgg_mf_chap./gau_chap;
54 img_rec=real(ifft2(img_rec_chap2)); img_rec2=img_rec(1:dimx,1:dimy);
55
56 seuil = .8; % 3e seuil
57 idx=find(abs(gau_chap) < seuil);
58 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
59 img_rec_chap3=imgg_mf_chap./gau_chap;
60 img_rec=real(ifft2(img_rec_chap3)); img_rec3=img_rec(1:dimx,1:dimy);

```

```

62 %% affichage filtrage inverse avec zero padding
63 figure(), colormap('bone')
64 subplot(2,5,1), imagesc(img), colorbar
65 title('original img'), axis('image')
66 subplot(2,5,6), imagesc(fx,fy,abs(fftshift(fft2(img))).^1), colorbar
67 title('|FFT(img)|^1'), axis('image')
68 subplot(2,5,2), imagesc(imgg(1:dim,1:dim)), colorbar
69 title('blurry+noisy img'), axis('image')
70 subplot(2,5,7), imagesc(fx,fy,abs(fftshift(imgg_chap)).^1), colorbar
71 title('|FFT(blurred img)|^1'), axis('image')
72
73 subplot(2,5,3), imagesc(img_rec1), colorbar
74 title('restaured img (seuil=0.1)'), axis('image')
75 subplot(2,5,8), imagesc(fx,fy,abs(fftshift(img_rec_chap1)).^1), colorbar
76 title('|FFT(rec'd img)|^1'), axis('image')
77
78 subplot(2,5,4), imagesc(img_rec2), colorbar
79 title('restaured img (seuil=0.5)'), axis('image')
80 subplot(2,5,9), imagesc(fx,fy,abs(fftshift(img_rec_chap2)).^1), colorbar
81 title('|FFT(rec'd img)|^1'), axis('image')
82
83 subplot(2,5,5), imagesc(img_rec3), colorbar
84 title('restaured img (seuil=0.8)'), axis('image')
85 subplot(2,5,10), imagesc(fx,fy,abs(fftshift(img_rec_chap3)).^1), colorbar
86 title('|FFT(rec'd img)|^1'), axis('image')

```

