

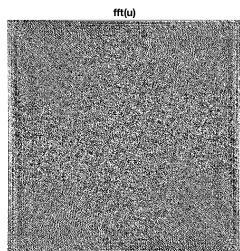
# TP 3 - Transformée de Fourier

Imagerie sous-pixelistique

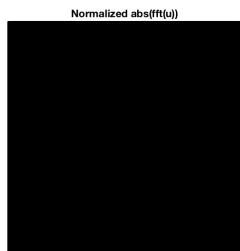
Tong ZHAO (tong.zhao@eleves.enpc.fr)

## 1 Exercice 7

On calcule tout d'abord la transformée de Fourier et on visualise la fonction originale, la valeur absolue de la fonction et la valeur absolue normalisée de la fonction.

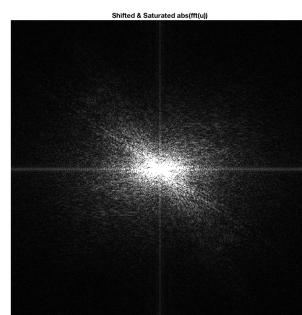
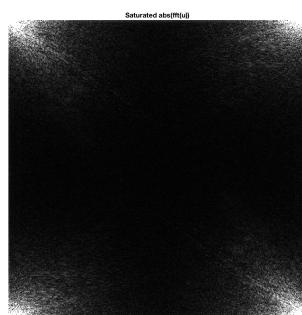


abs[fft(u)]



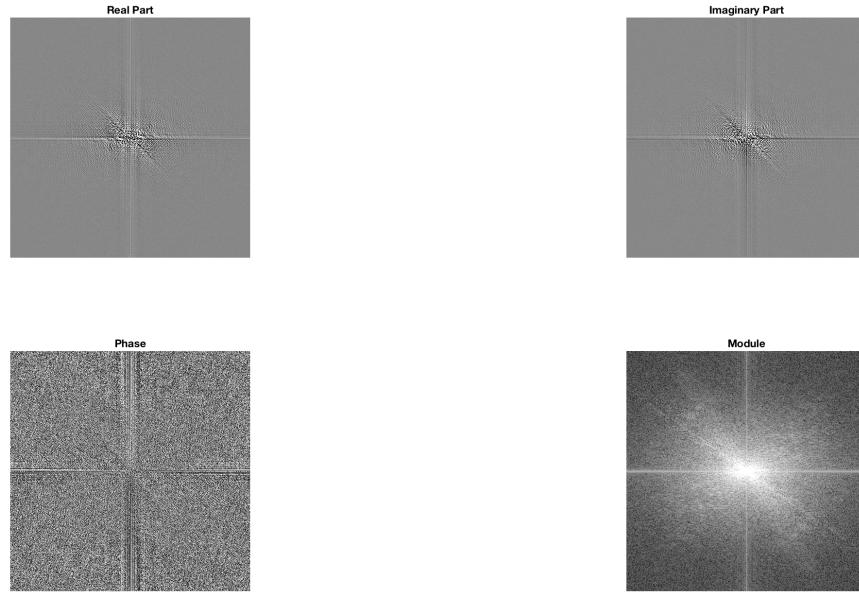
On voit que la fonction n'est pas centrée dans tous les trois figures. Tous les pixels dans la deuxième figure sont blancs, parce que la plupart des valeurs sont assez grandes. Tous les pixels deviennent noir après la normalisation parce que l'échelle des valeurs se varie beaucoup et la majorité des valeurs sont petites.

On applique la fonction *normsat* sur la tranformée de Fourier afin de saturer les 1% des valeurs les plus grandes.



La figure à droite est la fonction centrée. Après saturer les signaux de très hautes fréquences (causés par les bruits), on peut mieux visualiser la fonction.

On visualise la partie réelle, la partie imaginaire, la phase et le module de la transformée en échelle logarithmique et on obtient les résultats suivants.



## 2 Exercice 8

(1) On calcule la transformée de Fourier d'une image périodique.

$$\begin{aligned}
 \hat{U}(q, r) &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) \iint_{\mathbb{R}^2} \sum_{m, n \in \mathbb{Z}} \delta_{(k+mM, l+nN)}(x, y) e^{-i(qx+ry)} dx dy \\
 &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) \iint_{\mathbb{R}^2} \delta_{(mM, nN)}(x - k, y - l) e^{-i(qx+ry)} dx dy \\
 &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) e^{-i(kq+lr)} \iint_{\mathbb{R}^2} \delta_{(mM, nN)}(x, y) e^{-i(qx+ry)} dx dy \\
 &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) e^{-i(kq+lr)} \cdot \hat{\Pi}_{(M, N)}(q, r) \\
 &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) e^{-i(kq+lr)} \cdot \frac{2\pi}{N} \cdot \frac{2\pi}{M} \cdot \Pi_{(2\pi/M, 2\pi/N)}(q, r) \\
 &= \frac{4\pi^2}{MN} \sum_{(p, s) \in \mathbb{Z}^2} \hat{u}(p, s) \delta_{(2\pi p/M, 2\pi s/N)}(q, r)
 \end{aligned}$$

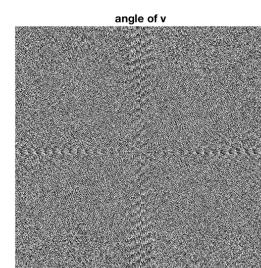
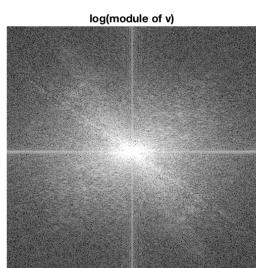
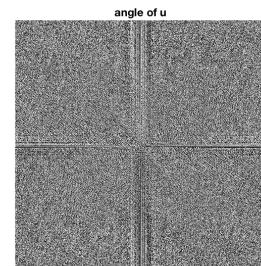
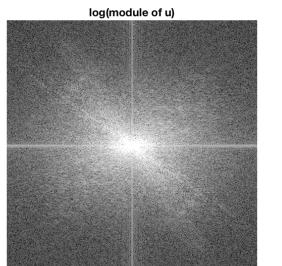
(2) On calcule la transformée de Fourier de  $\hat{u}'$ .

$$\begin{aligned}
\widehat{u'(k, l)}(q, r) &= \widehat{\dot{u}(k + k_0, l + l_0)}(q, r) \\
&= \widehat{\dot{u}(k, l)}(q, r) \cdot \exp\left(-2\pi i\left(\frac{qk}{M} + \frac{rl}{N}\right)\right) \\
&= \widehat{u(k, l)}(q, r) \cdot \exp\left(-2\pi i\left(\frac{qk}{M} + \frac{rl}{N}\right)\right)
\end{aligned}$$

(3) On visualise l'image originale  $u$  et l'image translatée  $v$ .



On compare ensuite le module et la phase de leurs transformées de Fourier. On observe bien que les modules sont les mêmes mais la phase change à cause de la translation, ce qui correspond à la formule au-dessus.



### 3 Exercice 9

On applique l'interpolée de Shannon sur une image discrète  $u$  de taille  $M \times N$ , où  $M$  et  $N$  sont pairs.

$$\begin{aligned}
v(x, y) &= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) \cdot sincd_M(x - k) \cdot sincd_N(y - l) \\
&= \frac{1}{MN} \sum_{|\alpha| \leq \frac{M}{2}, \alpha \in \mathbb{Z}} \sum_{|\beta| \leq \frac{N}{2}, \beta \in \mathbb{Z}} \hat{u}[\alpha, \beta] \epsilon_M(\alpha) \epsilon_N(\beta) e^{2\pi i (\frac{x\alpha}{M} + \frac{y\beta}{N})} \\
&= \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) \cdot \frac{1}{MN} \sum_{|\alpha| \leq \frac{M}{2}, \alpha \in \mathbb{Z}} \sum_{|\beta| \leq \frac{N}{2}, \beta \in \mathbb{Z}} Re(e^{2\pi i \frac{\alpha(x-k)}{M}}) Re(e^{2\pi i \frac{\beta(y-l)}{N}})
\end{aligned}$$

On considère la fonction  $w$ :

$$w(x, y) = Re \left( \frac{1}{MN} \sum_{|\alpha| \leq \frac{M}{2}, \alpha \in \mathbb{Z}} \sum_{|\beta| \leq \frac{N}{2}, \beta \in \mathbb{Z}} \hat{u}(\alpha, \beta) e^{2\pi i (\frac{x\alpha}{M} + \frac{y\beta}{N})} \right)$$

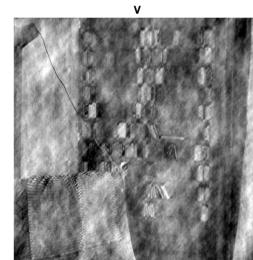
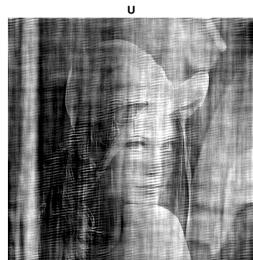
Le résidu  $w - v$  est donc:

$$w(x, y) - v(x, y) = \sum_{0 \leq k \leq M-1, 0 \leq l \leq N-1} u(k, l) \cdot \frac{1}{MN} \sum_{|\alpha| \leq \frac{M}{2}, \alpha \in \mathbb{Z}} \sum_{|\beta| \leq \frac{N}{2}, \beta \in \mathbb{Z}} Im(e^{2\pi i \frac{\beta(x-k)}{M}}) Im(e^{2\pi i \frac{\beta(y-l)}{N}})$$

L'interpolation par  $v$  est la plus satisfaisante du point de vue des invariances géométriques.

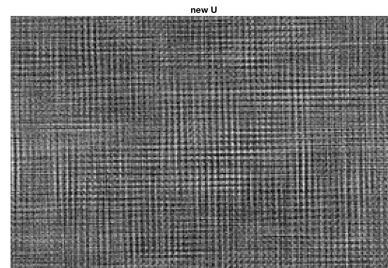
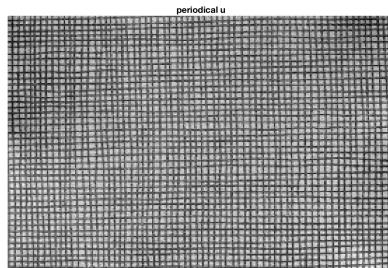
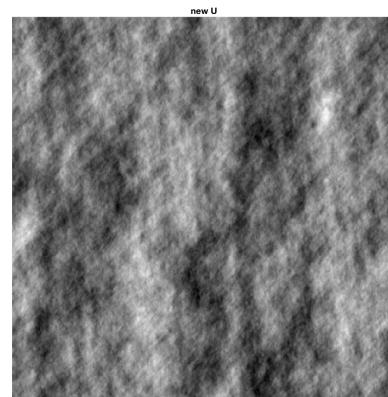
## 4 Exercice 10

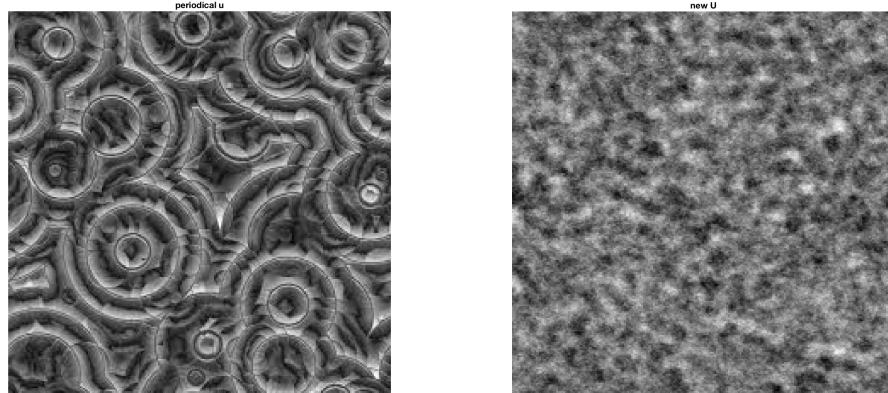
(1) On implémente la fonction *exchange\_phase* et on l'applique sur deux images.



Après on change des phases entre les deux images, on voit que les images sont bruitées. Mais on peut arriver à reconnaître l'image de départ quand on conserve son module.

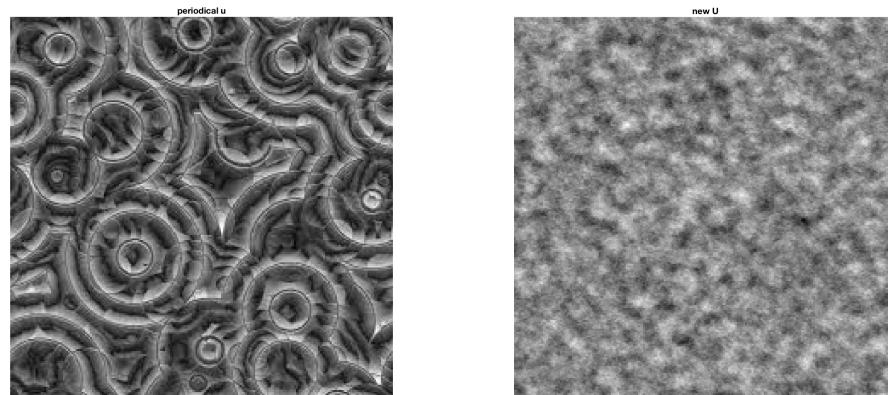
(2) Ici on visualise l'opération sur trois images: l'image *lena* (image classique), *circle* (image de texture) et *square* (image de texture).



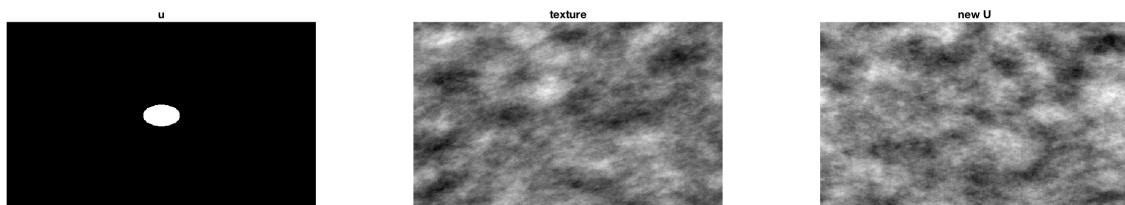


On voit clairement que les images de texture sont mieux reproduit. L'argument est que la plupart du texture est codé dans le module de l'DFT de l'image et la plupart du contour géométrique est codé dans sa phase.

Si on travaille directement avec l'image de départ, le texture sera moins évident. Ici on montre ce phénomène sur l'image *circle*.



**(3)** On essaye plusieurs paramètres et on trouve que quand  $a = 1, b = 1/\sqrt{3}$ , le texture est similaire comme celui de *texture.pgm*.



**Fiche 1: exchange\_phase.m**

```
1      function [ U, V ] = exchange_phase( u, v )
2      %change the module and the phase of two images.
3
4          fu = fft2(u);
5          fv = fft2(v);
6
7          absu = abs(fu);
8          absv = abs(fv);
9
10         phaseu = angle(fu);
11         phasev = angle(fv);
12
13         fU = absv .* exp(1i * phaseu);
14         fV = absu .* exp(1i * phasev);
15
16         U = real(ifft2(fU));
17         V = real(ifft2(fV));
18
19     end
```

**Fiche 2: main.m**

```
1
2      %% Exercice 7
3
4      u = double(imread('images/lena.pgm'));
5      f = fft2(u);
6      figure(1);
7      subplot(2, 2, 1);
8      imshow(f);
9      title('fft(u)');
10     subplot(2, 2, 2);
11     imshow(abs(f));
12     title('abs(fft(u))');
13     subplot(2, 2, 3);
14     imshow(abs(f),[]);
15     title('Normalized abs(fft(u))');
16     figure(2);
17     subplot(1, 2, 1);
18     imshow(normsat(abs(f),1));
19     title('Saturated abs(fft(u))')
20     subplot(1, 2, 2);
21     imshow(normsat(fftsat(abs(f)),1));
22     title('Shifted & Saturated abs(fft(u))')
23
24     figure(3);
25     subplot(2, 2, 1);
```

```

26 imshow(normsat(fftshift(real(f)), 1));
27 title('Real Part');
28 subplot(2, 2, 2);
29 imshow(normsat(fftshift(imag(f)), 1));
30 title('Imaginary Part');
31 subplot(2, 2, 3);
32 imshow(normsat(fftshift(angle(f)), 1));
33 title('Phase');
34 subplot(2, 2, 4);
35 imshow(normsat(fftshift(log(abs(f))), 1));
36 title('Module');

37 %% Exercice 8

38 %% Q3

39
40 u = double(imread('lena.pgm'));
41 figure(2);
42 subplot(1, 2, 1);
43 imshow(u, []);
44 title('u');
45 v = fshift(u, -30, -30);
46 subplot(1, 2, 2);
47 imshow(v, []);
48 title('v');

49 fu = fft2(u);
50 fv = fft2(v);
51 figure(3);
52 subplot(2, 2, 1);
53 imshow(normsat(fftshift(log(abs(fu))), 1));
54 title('log(module of u)');
55 subplot(2, 2, 2);
56 imshow(normsat(fftshift(angle(fu)), 1));
57 title('angle of u');
58 subplot(2, 2, 3);
59 imshow(normsat(fftshift(log(abs(fv))), 1));
60 title('log(module of v)');
61 subplot(2, 2, 4);
62 imshow(normsat(fftshift(angle(fv)), 1));
63 title('angle of v');

64 %% Exercice 10

65 %% Q1
66 u = double(imread('images/lena.pgm'))/255;
67 v = double(imread('images/room.pgm'))/255;

```

```

74 [U, V] = exchange_phase(u, v);
75
76 figure(4);
77 subplot(2, 2, 1);
78 imshow(u, []);
79 title('u');
80 subplot(2, 2, 2);
81 imshow(v, []);
82 title('v');
83 subplot(2, 2, 3);
84 imshow(U);
85 title('U');
86 subplot(2, 2, 4);
87 imshow(V);
88 title('V');

89 %% Q2
90 u = imread('images/circle.jpeg');
91 if prod(size(size(u))) == 3
92     u = double(rgb2gray(u)) / 255;
93 else
94     u = double(u) / 255;
95 end
96 p_perde = perdecomp(u);
97 phasep = randphase(u);

98 figure(5);
99 subplot(1, 2, 1);
100 imshow(p_perde, []);
101 title('periodical u');
102 subplot(1, 2, 2);
103 imshow(phasep, []);
104 title('new U');

105 %% Q3
106
107 v = double(imread('images/texture.pgm')) / 255;
108
109 w = 500;
110 h = 300;
111 Iw = -w/2:-w/2+w-1;
112 Ih = -h/2:-h/2+h-1;
113 [X, Y] = meshgrid(Iw, Ih);
114 R = sqrt(1 * X.^2 + 3 * Y.^2);
115 u = (R<30);
116 randu = randphase(u);

117 figure(6);

```

```
122 subplot(1, 3, 1);
123 imshow(u, []);
124 title('u');
125 subplot(1, 3, 2);
126 imshow(v, []);
127 title('texture');
128 subplot(1, 3, 3);
129 imshow(randu, []);
130 title('new U');
```