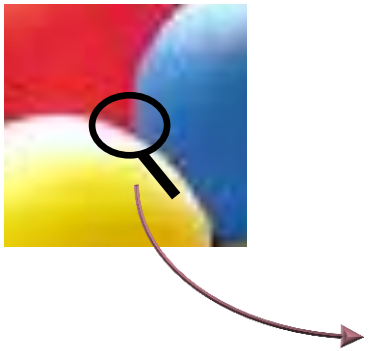
ENSC - 2ème année

Initiation au Traitement des Images

Plan

- Image couleur
  - Lecture/Ecriture/Synthèse
  - Visualisation
- Espaces caractéristiques
  - Couleur/Amplitude
  - Spatial
  - Fréquentiel
- Traitements
  - Histogramme
  - Filtrage/Détection de contours

Définition (1 /2)



R=212 G=16 B=40	R=205 G=65 B=112	R=103 G=120 B=176	R=62 G=127 B=193
R=201 G=26 B=43	R=197 G=69 B=94	R=154 G=106 B=148	R=98 G=117 B=186
R=192 G=101 B=106	R=138 G=59 B=80	R=127 G=96 B=137	R=97 G=129 B=188
R=255 G=250 B=250	R=230 G=192 B=213	R=140 G=118 B=156	R=73 G=97 B=145
R=250 G=248 B=251	R=255 G=248 B=255	R=255 G=246 B=255	R=182 G=176 B=210

Intensité vectorielle

40	112	176	193
43	94	148	186
106	80	137	188
250	213	156	145
251	255	255	210

Matrice B

⊕

16	65	120	127
26	69	106	117
101	59	96	129
250	192	118	97
248	248	246	176

Matrice V

« Vraies » couleurs

- Composante rouge (R)
- Composante verte (V)
- Composante bleue (B)

212	205	103	62
201	197	154	98
192	138	127	97
255	230	140	73
250	255	255	182

Matrice R

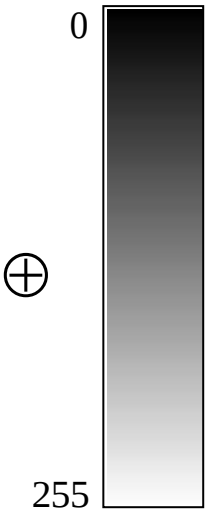
⊕

Format usuel : entiers 0 à 255

Définition (2/2)

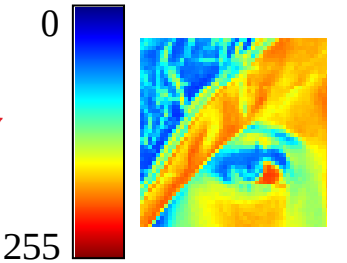
60	64	62	62	55	51
59	73	98	90	66	54
72	105	167	170	120	74
88	91	188	202	184	150
103	77	191	205	203	190
75	131	208	209	207	202

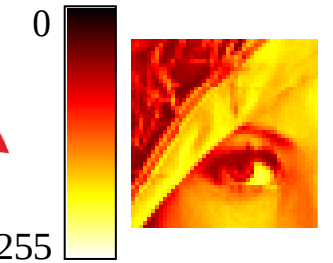
Intensité scalaire

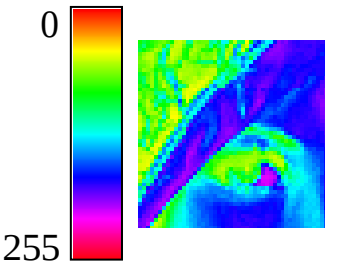


Palette
(niveaux de gris)

Autres palettes







Couleurs indexées ou « fausses » couleurs

- Scalaire (→index de table)
- Palette (table de correspondance couleur)

Fichiers en « vraie couleur »

```
>> A=imread('pool.tif');
>> whos
  Name        Size                          Bytes  Class

  A         383x510x3                      585990  uint8 array

Grand total is 585990 elements using 585990 bytes

>> figure, image(A)
>> A(:,:,1)
>> A(:,:,2)
>> A(:,:,3)
```

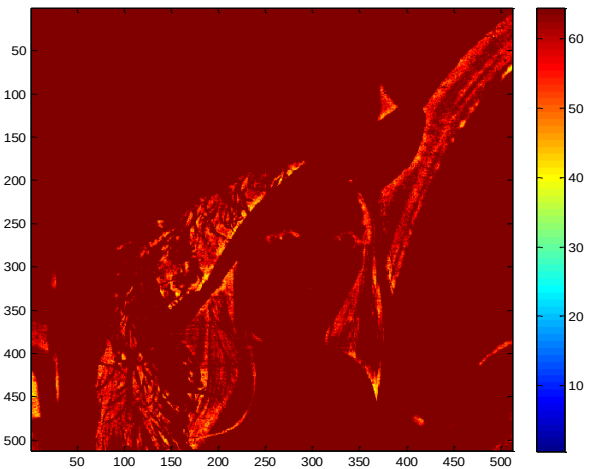
hauteur

largeur

« vraies » couleurs

Fichiers en couleur indexée

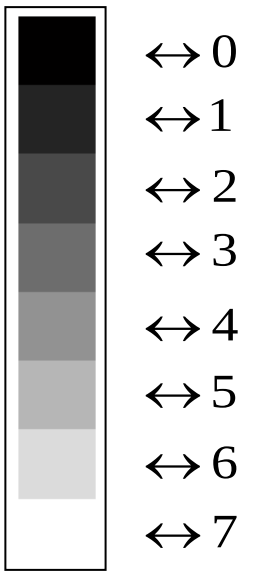
```
>> A=imread('lena.bmp');
>> whos
  Name        Size                     Bytes  Class

  A         512x512                   262144  uint8 array

Grand total is 262144 elements using 262144 bytes

>> figure, image(A), colorbar
>> A
>> figure, image(A), colormap(gray(256)), colorbar
```



Palette ?

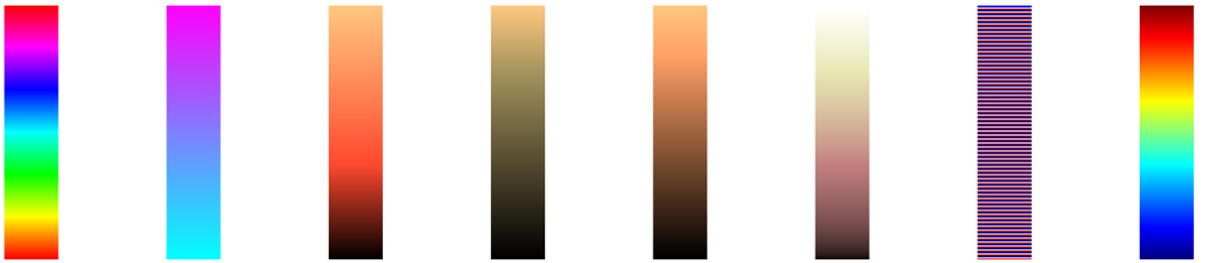
```
>> gray(8)

ans =
         0         0         0
    0.1429    0.1429    0.1429
    0.2857    0.2857    0.2857
    0.4286    0.4286    0.4286
    0.5714    0.5714    0.5714
    0.7143    0.7143    0.7143
    0.8571    0.8571    0.8571
    1.0000    1.0000    1.0000
```

R, G, B



Autres palettes : hsv, cool, hot, bone, copper, pink, flag, jet, …

64 niveaux par défaut



Combien de couleurs ?

Notion de profondeur

nombre (fini) de bits codant la couleur de chaque pixel (acquisition, stockage ou affichage)

→ fonction Matlab : **get(0,'ScreenDepth')**

Valeurs usuelles

- « Vraies » couleurs
  - ✓ 16 bits (1+3*5 soit 32768 couleurs)
  - ✓ 24 bits (3*8 soit 16 777 216 couleurs) ou 32 bits (3*8+1*8, canal alpha) : cartes graphiques
  - ✓ 30 bits ou 48 bits : numériseurs
- Couleurs indexées
  - ✓ 1 bit (noir/blanc, noir/vert, noir/orange, …) : écran de production
  - ✓ 8 bits (palette de 256 couleurs) : stockage
  - ✓ 16 bits (palette de 65536 couleurs) : smartphones

Fichiers : lecture/écriture

imread/imwrite

JPEG Any baseline JPEG image; JPEG images with some commonly used extensions; 8-bit and 12-bit lossy compressed RGB and grayscale images; 8-bit and 12-bit lossless compressed RGB images; 8-bit, 12-bit, and 16-bit lossless compressed grayscale images

TIFF Any baseline TIFF image, including 1-bit, 8-bit, and 24-bit uncompressed images; 1-bit, 8-bit, and 24-bit images with packbits compression; 1-bit images with CCITT compression; 12-bit grayscale, 12-bit indexed, and 36-bit RGB images; 16-bit grayscale, 16-bit indexed, and 48-bit RGB images; 24-bit and 48-bit ICCLAB and CIELAB images; 32-bit and 64-bit CMYK images; and 8-bit tiled TIFF images with any compression and colorspace combination listed above.

GIF Any 1-bit to 8-bit GIF image

BMP 1-bit, 4-bit, 8-bit, 16-bit, 24-bit, and 32-bit uncompressed images; 4-bit and 8-bit run-length encoded (RLE) images

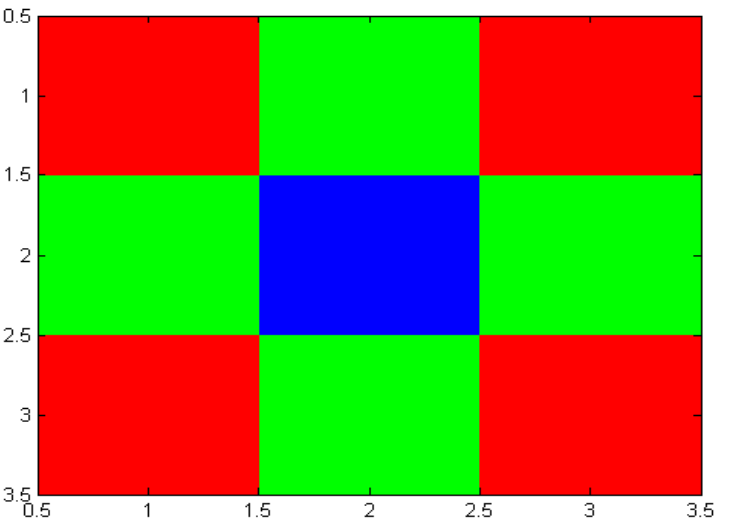
PNG Any PNG image, including 1-bit, 2-bit, 4-bit, 8-bit, and 16-bit grayscale images; 8-bit and 16-bit indexed images; 24-bit and 48-bit RGB images

Fichier et palette

```
>> [A,map]=imread('lena.bmp');
>> figure, image(A), colormap(map), colorbar
```

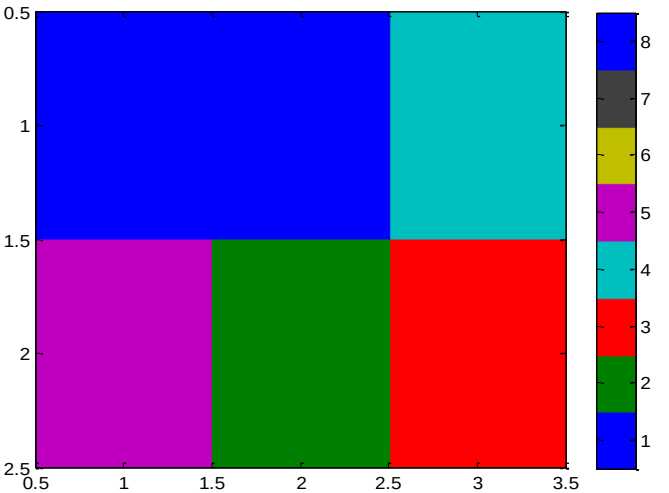
Synthèse en « vraies couleurs »

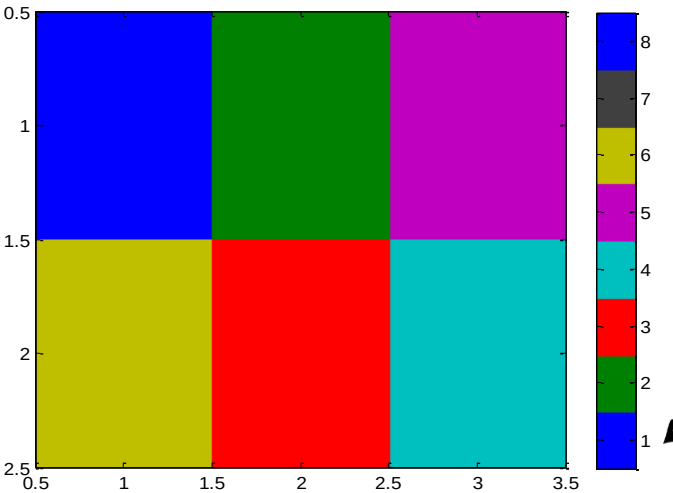
```
clear all
close all

r=[1 0 1;0 0 0;1 0 1];
g=[0 1 0;1 0 1;0 1 0];
b=[0 0 0;0 1 0;0 0 0];

img=cat(3,r,g,b);
figure, image(img)
```



Synthèse en couleurs indexées

```
clear all
close all

img=[0 1 4;5 2 3];
figure, image(img)
colormap(lines(8)), colorbar



figure, image(uint8(img))
colormap(lines(8)), colorbar

figure, image(img+1)
colormap(lines(8)), colorbar
```





0 ou 1 !!!

Fonctions Matlab de visualisation

	image	imagesc	imshow
Formats	uint8, uint16, double	uint8, uint16, double	uint8, uint16, single, double, …
Palette initiale	jet (64 niveaux)	jet (64 niveaux)	gray (256 niveaux)
Intervalles (couleurs indexées)	[0,N-1] uint8, uint16 [1,N] double (sur N couleurs)	[0,N-1] uint8, uint16 [1,N] double (sur N couleurs)	[0,255] uint8 [0,65535] uint16 [0,1] single/double
Intervalles (« vraies » couleurs)	[0,255] uint8, uint16 [0,1] double	[0,255] uint8, uint16 [0,1] double	[0,255] uint8 [0,65535] uint16 [0,1] single, double
Grossissement initial	variable	variable	1
Ratio L/H initial	variable	variable	1 (pixels carrés)
Divers		Mise à l'échelle des intensités (auto. par défaut/ contrôlée [.])	Toolbox « Image Processing »

Problématiques

- Souci de fidélité (respect des intensités initiales)
  - **image/imshow**
- Souci « informationnel » (couleurs indexées uniquement)
  - Etalement automatique des intensités (**imagesc**)
  - Etalement contrôlé des intensités (**imagesc/imshow(…,[ ])**)
- Souci de comparaison (couleurs indexées uniquement)
  - Etalement identique des intensités (**imagesc/imshow(…,[ ])**)
- Choix d'une palette (couleurs indexées uniquement)
  - **colormap** (**image**/**imagesc** mais pas **imshow** !!!)
  - **imshow(…, map)**

Formats et opérations

```
>> clear all
>> close all
>> A=imread('lena.bmp');
>> B=zeros(512);
>> B(250:370,240:350)=300;
>> C=A+B;
??? Error using ==> plus
Integers can only be combined with
integers of the same class, or scalar
doubles.

>> C=double(A)+B;
>> image(C)
>> colormap(gray(256))
```

Comparaison image/imagesc

```
clear all
close all

A=imread('lena.bmp');
figure, image(A)
colormap(gray(16))
colorbar
figure, imagesc(A)
colormap(gray(16))
colorbar

A=imread('lezard.bmp');
figure, imagesc(A, [0
80])
colormap(gray(256))
colorbar
```

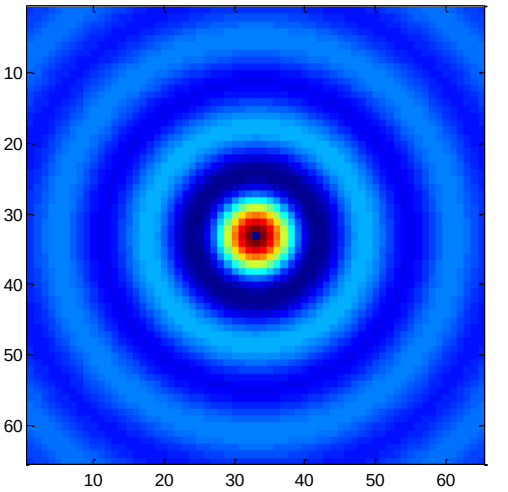
```
clear all
close all

A=double(imread('coronair.tif'));
BW=A>150;
B=A.*not(BW)+500.*BW;
figure
image(B)
colormap(gray(256))
colorbar
figure
imagesc(B)
colormap(gray(256))
colorbar
```

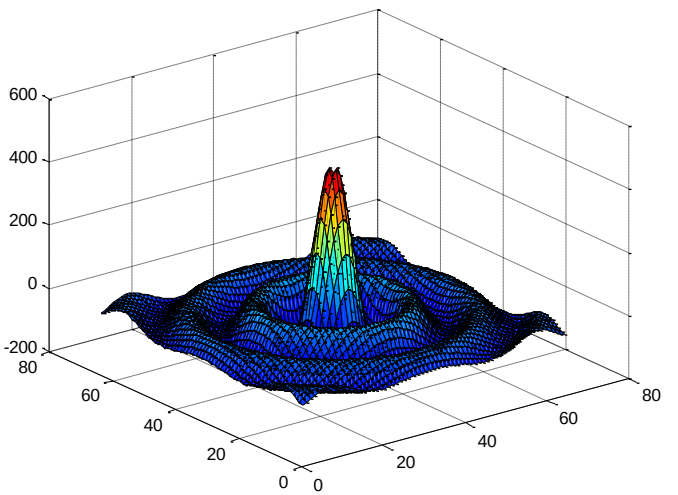
Fonction imshow

```
clear all
close all

A=imread('cameraman.tif');
figure
imshow(A)

B=double(A);
figure
imshow(B)

figure
imshow(B/255)
```

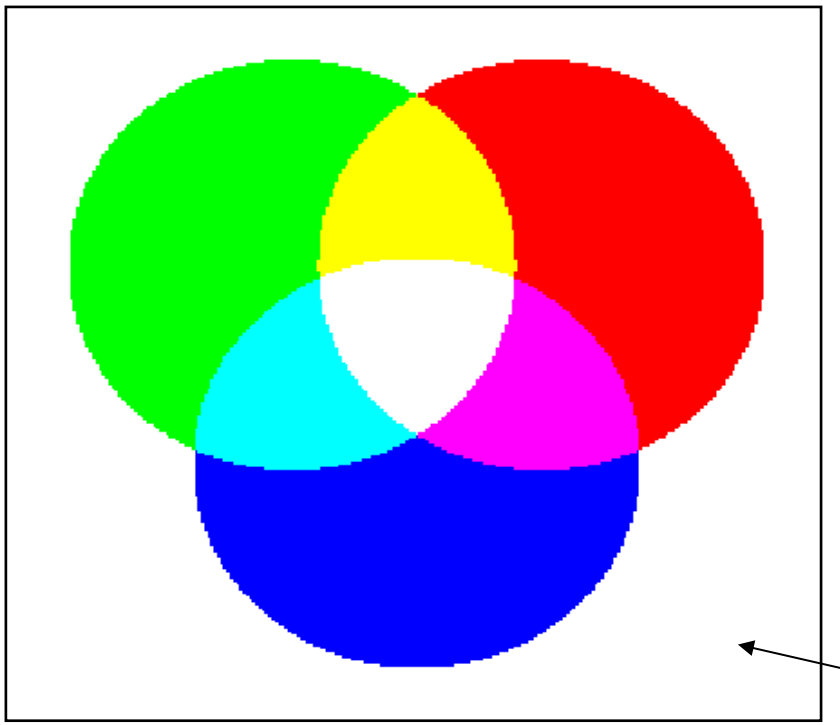
Synthèse analytique : meshgrid

```
clear all
close all

[X,Y]=meshgrid(-32:32,-32:32);
R=(X.^2+Y.^2).^0.5;

img=1000*sin(R/2)./R;
figure, imagesc(img), axis square
figure, surf(img)
```





Exercice : synthèse additive



- « Vraies » couleurs
- Couleurs indexées

Exercice : vidéo

- Synthèse additive dynamique (couleurs circulantes)

  99 images, 5 images/seconde, sans compression

- **avifile, addframe, close**

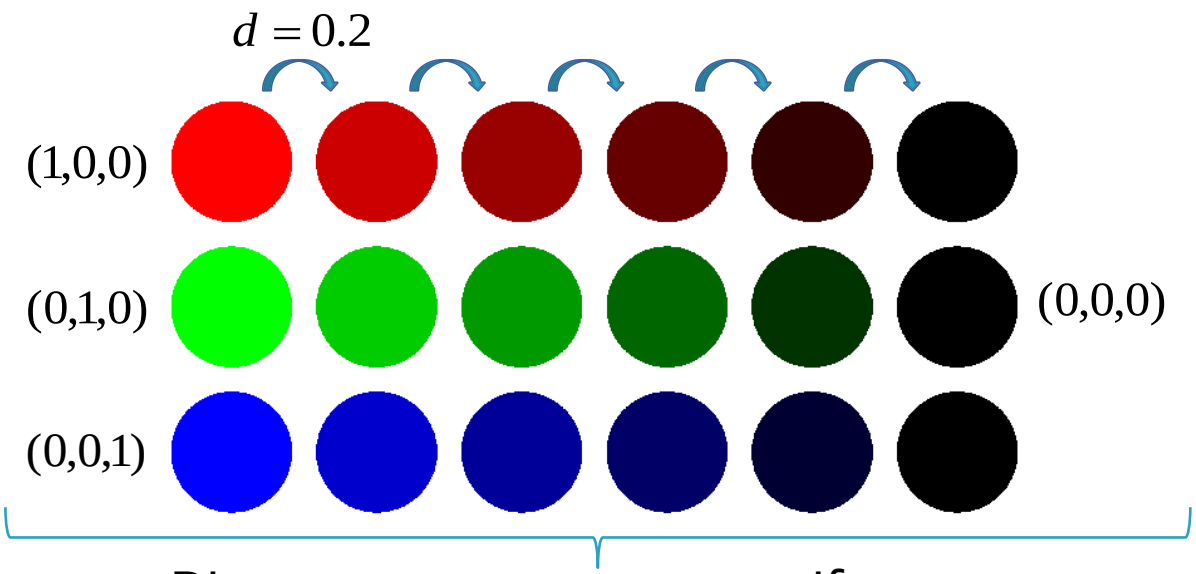
Exercice : interaction

- Affichage d'une image
- Capture d'un segment (ginput), de gauche à droite et de haut en bas
- Extraction du profil (signal horizontal ou vertical le plus « proche ») correspondant
- Affichage du profil

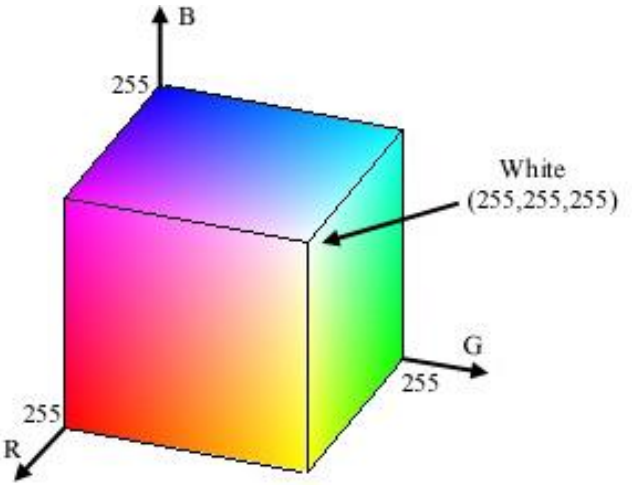
Espaces caractéristiques

- Couleur/Amplitude
- Spatial
- Fréquentiel

Espace chromatique RVB(RGB)



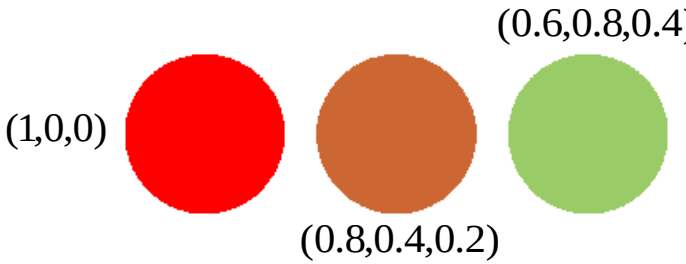
Distances perçues non « uniformes »



$$L = \frac{R+V+B}{3}$$

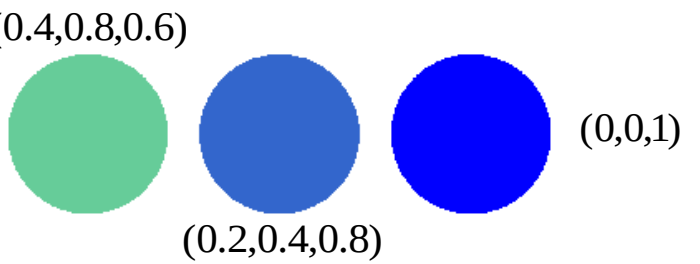
luminance (intensité moyenne)

Espace d'acquisition du système visuel humain



R, G, B ?

décomposition très difficile (même à 20% près)



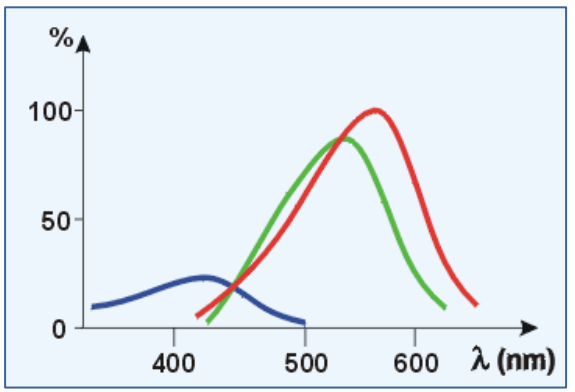
Luminance vs Chrominance

luminosité plus forte

Niveaux de gris

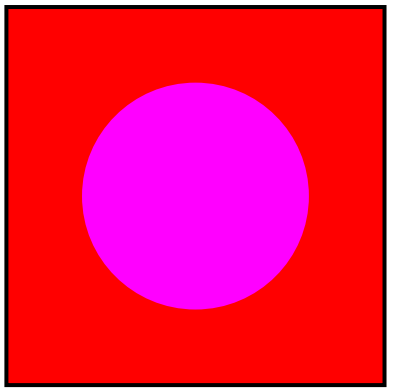
contrastes plus faibles

Palettes de primaires pures
(intensités identiques et autres composantes éteintes)

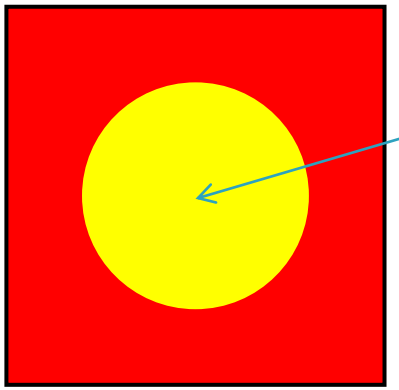


Sensibilité relative des photo-récepteurs de chrominance

Luminance « perceptuelle »

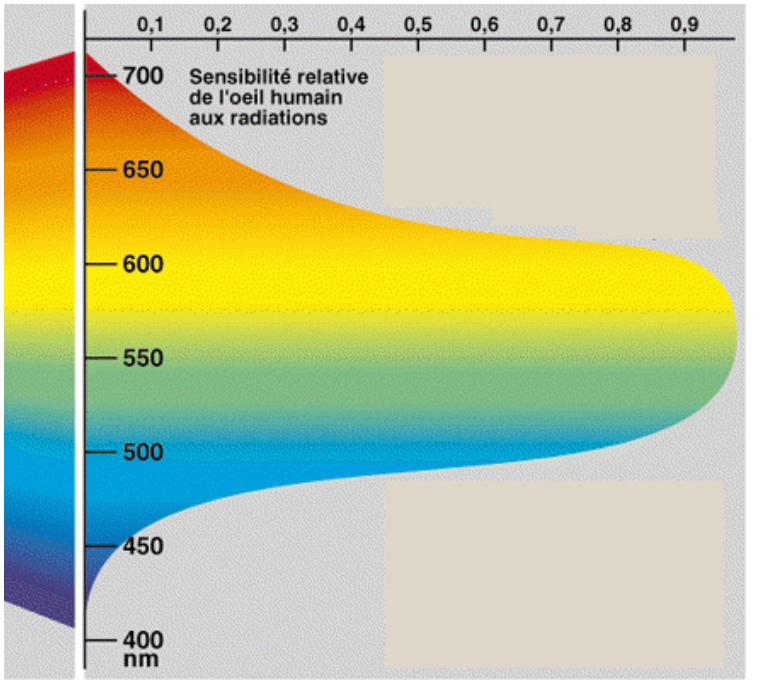


Rouge/Bleu



luminosité plus forte

Rouge/Vert

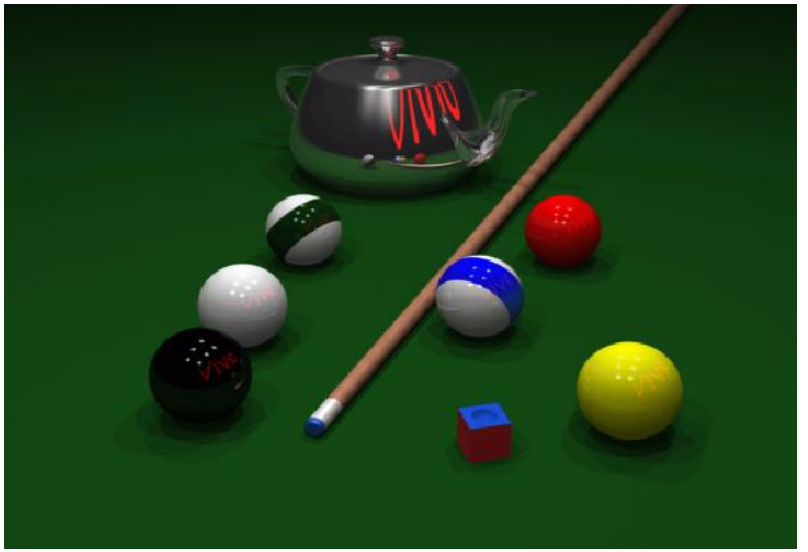


$$Y = 0.299\,R + 0.587\,V + 0.114\,B$$

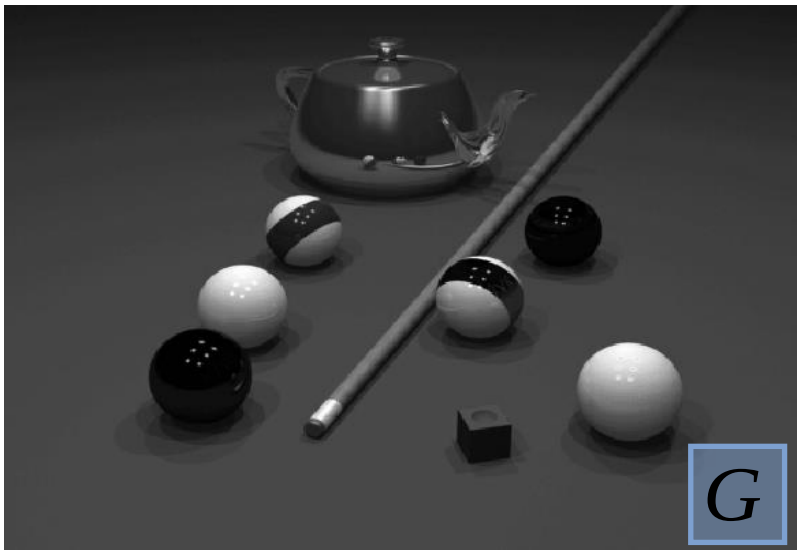
pondération plus faible

Espace colorimétrique YCbCr

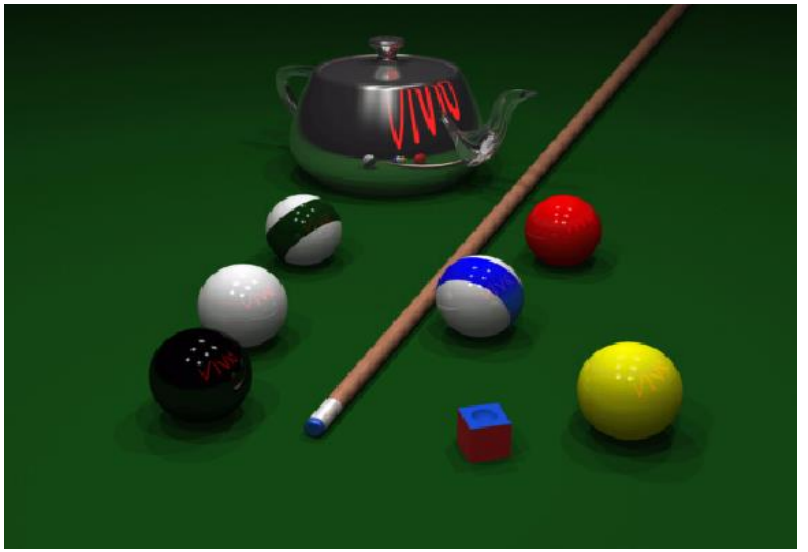
```
clear all
close all

A=double(imread('pool.tif'));
R=A(:,:,1);
G=A(:,:,2);
B=A(:,:,3);

Y=0.299*R+0.587*G+0.114*B;
Cb=0.564*(B-Y)+128;
Cr=0.713*(R-Y)+128;

figure, imshow(uint8(A))
figure, imshow(uint8(R))
figure, imshow(uint8(G))
figure, imshow(uint8(B))
figure, imshow(uint8(Y))
figure, imshow(uint8(Cb))
figure, imshow(uint8(Cr))
```



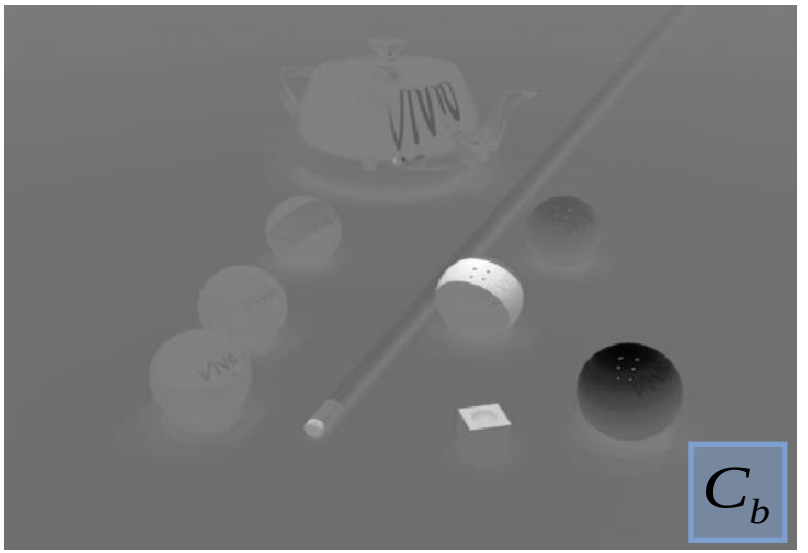

R

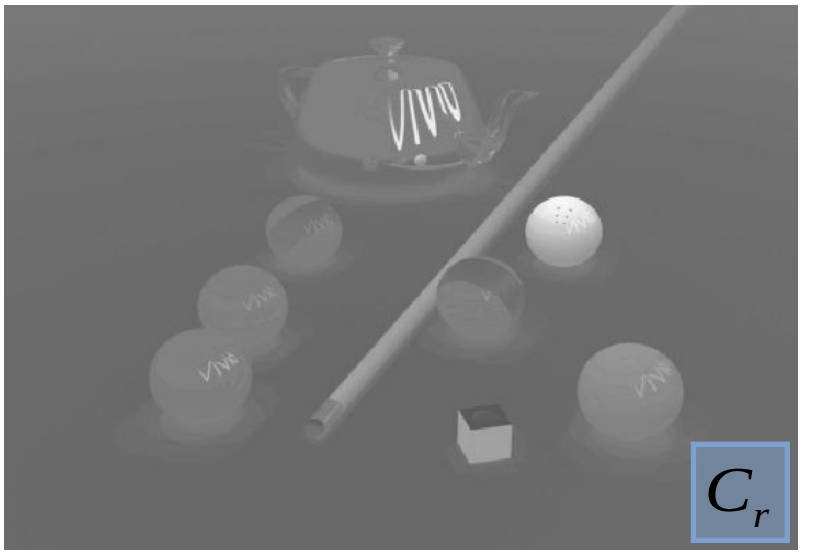

G

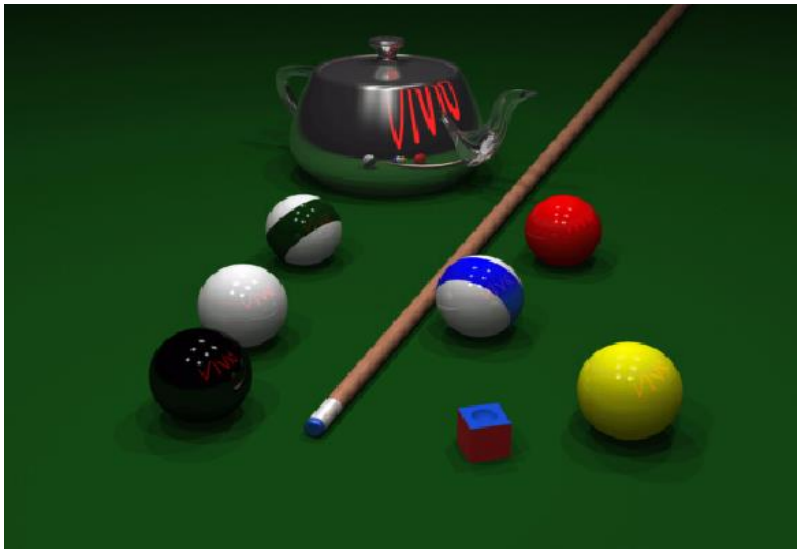

B

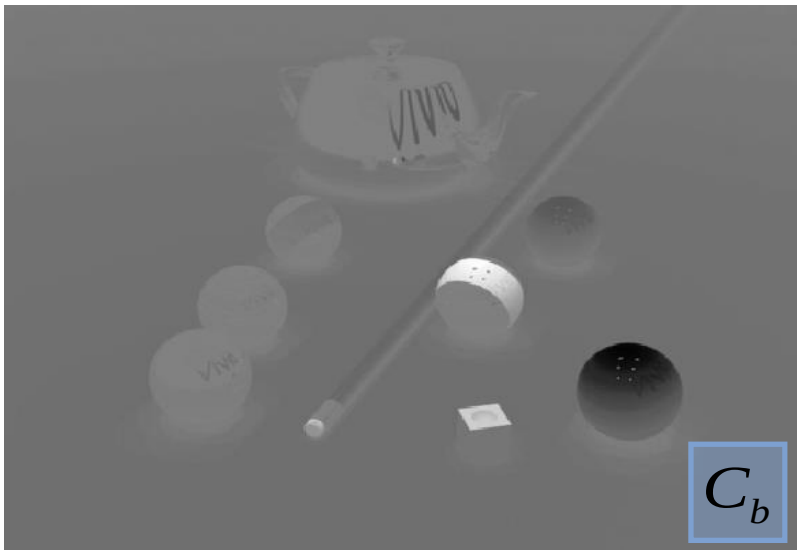



Y


C_b


C_r

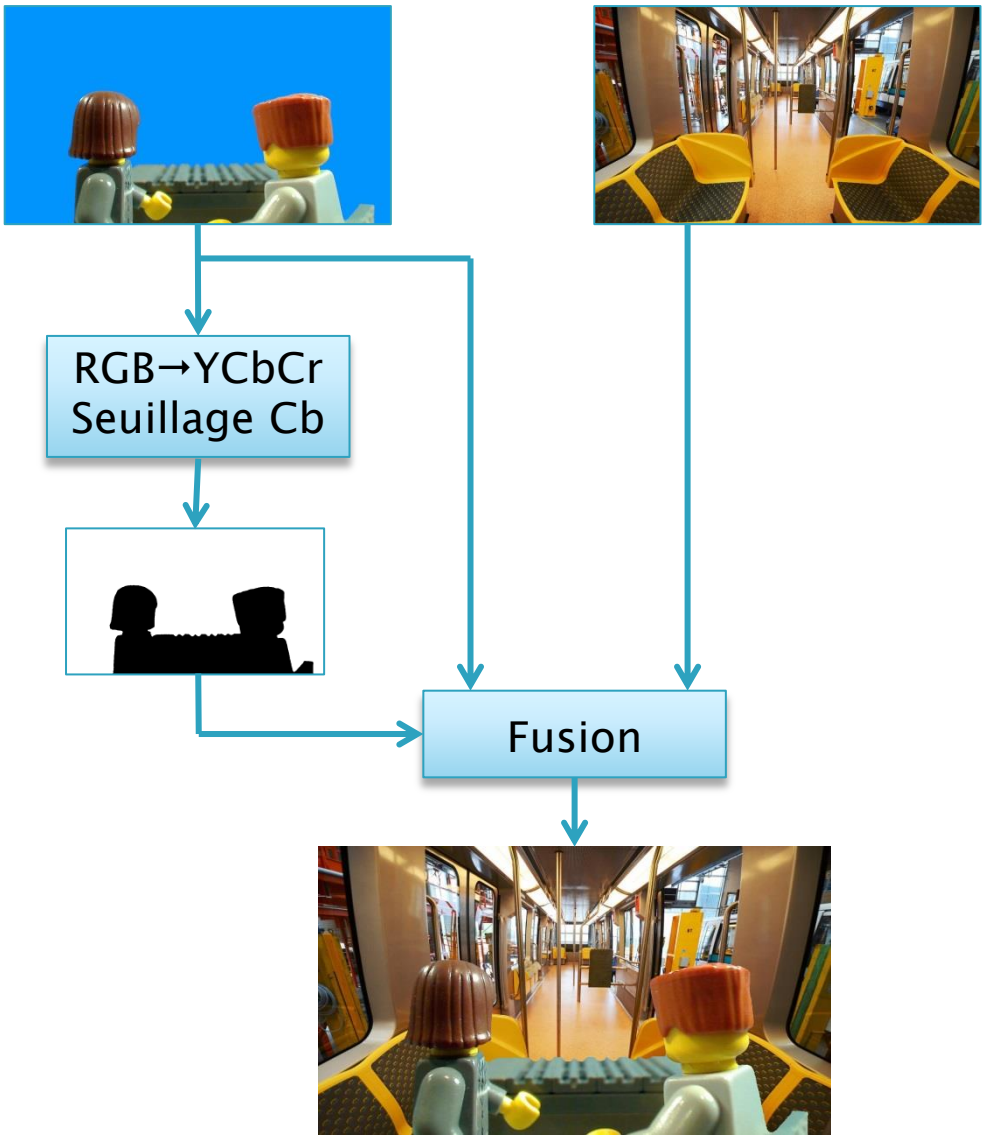



C_b

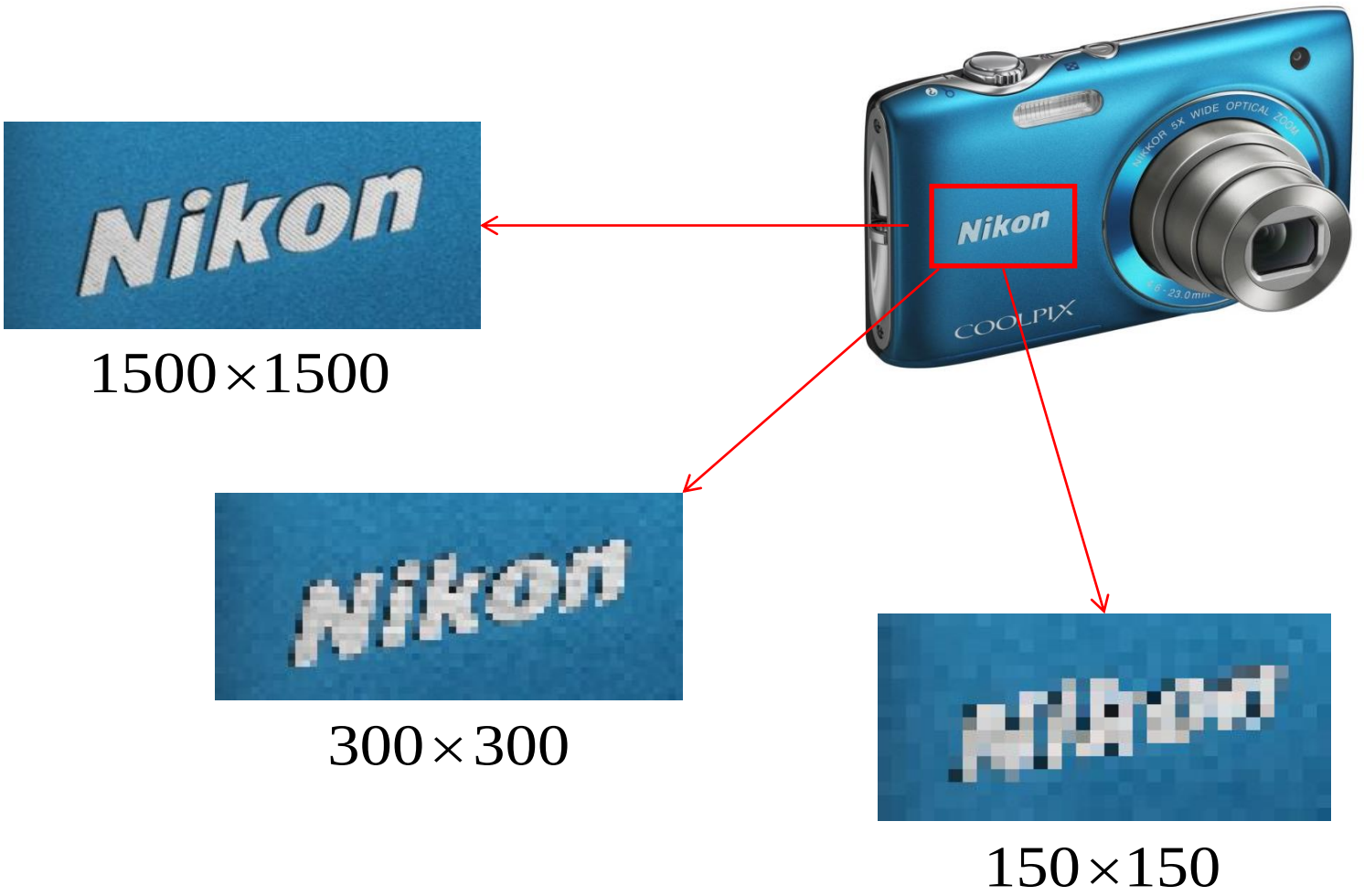

B

Chroma-Keying

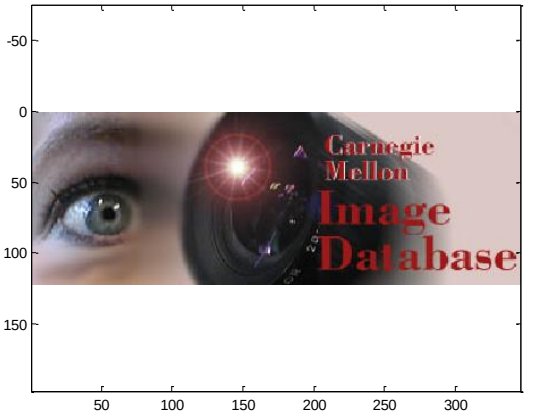
```
clear all
close all

img1=double(imread('people.jpg'));
R1=img1(:,:,1);
G1=img1(:,:,2);
B1=img1(:,:,3);
Y1=0.299*R1+0.587*G1+0.114*B1;
Cb1=0.564*(B1-Y1)+128;
M=double(Cb1>150);

img2=double(imread('metro.jpg'));
R2=img2(:,:,1);
G2=img2(:,:,2);
B2=img2(:,:,3);

R2=R1.*(1-M)+R2.*M;
G2=G1.*(1-M)+G2.*M;
B2=B1.*(1-M)+B2.*M;
img = uint8(cat(3,R2,G2,B2));
figure, imshow(img)
```

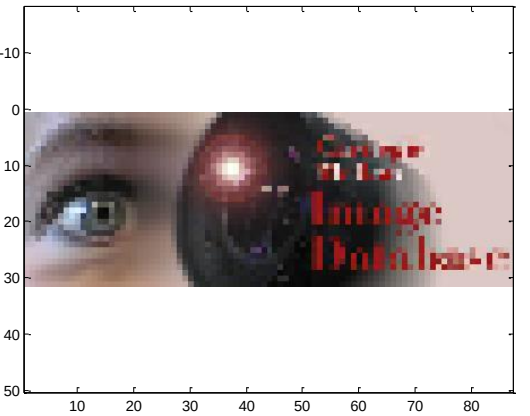


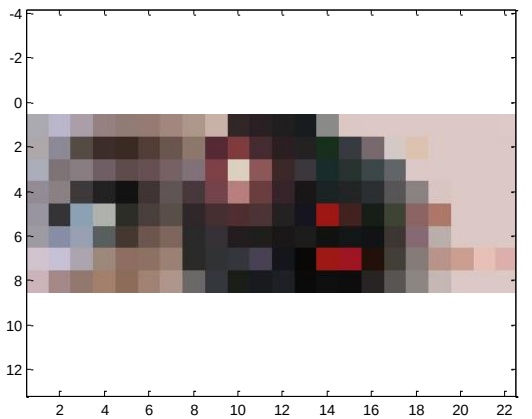
Résolution spatiale



```
clear all
close all

A=imread('title.jpg');
figure
image(A)
axis('equal')
[Ny,Nx,Nz]=size(A);

figure
image(A(1:4:Ny,1:4:Nx,:))
axis('equal')
figure
image(A(1:16:Ny,1:16:Nx,:))
axis('equal')
```

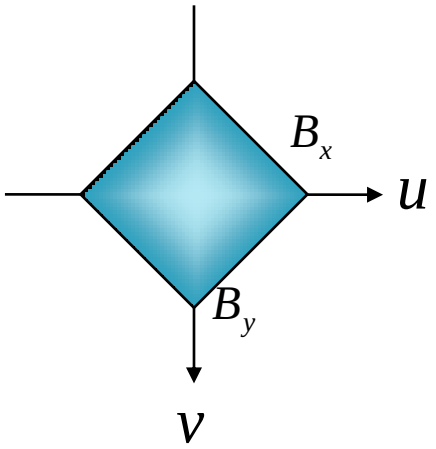




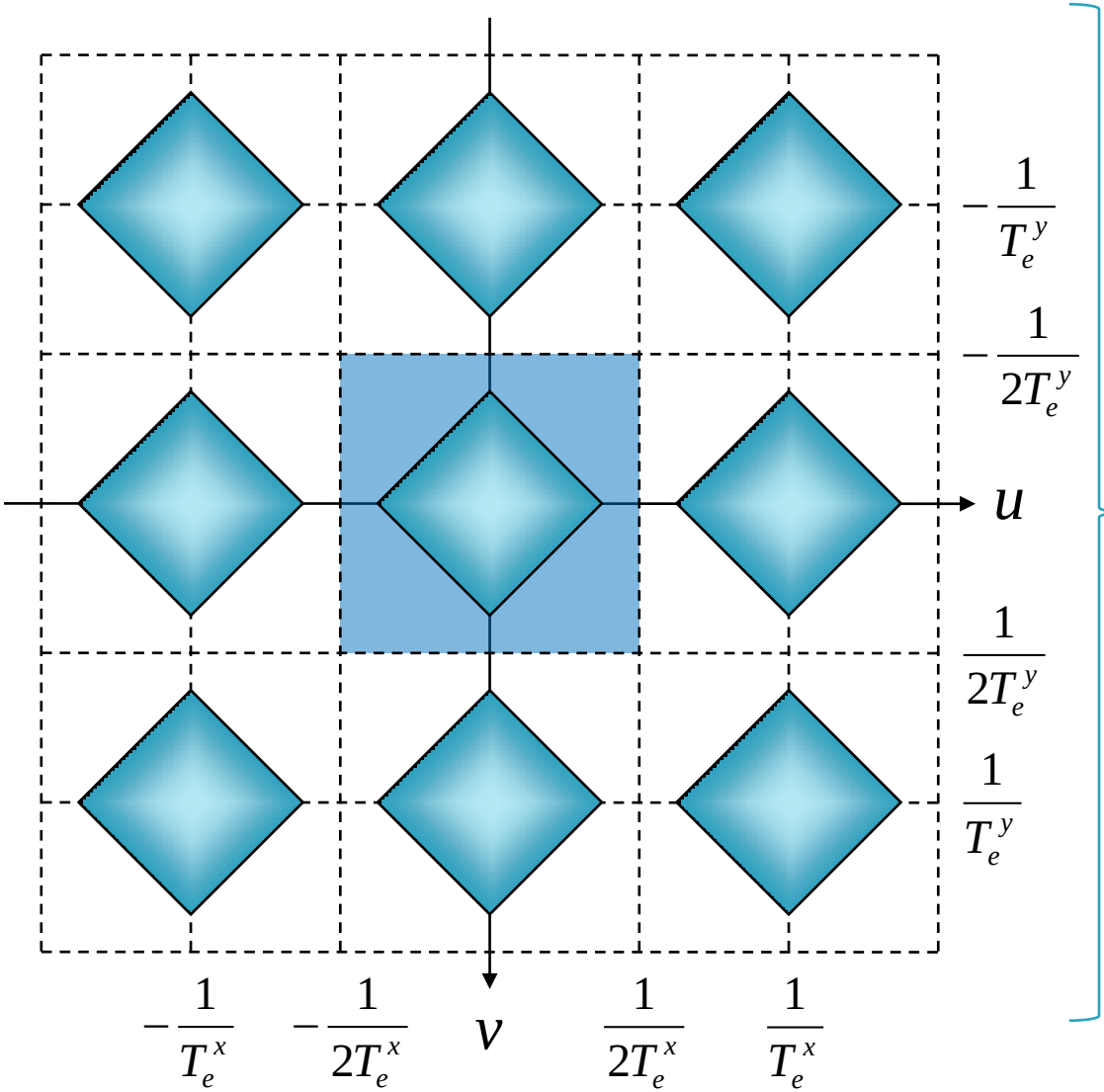


Spectre d'une séquence discrète 2d

$|F_a(u)|$



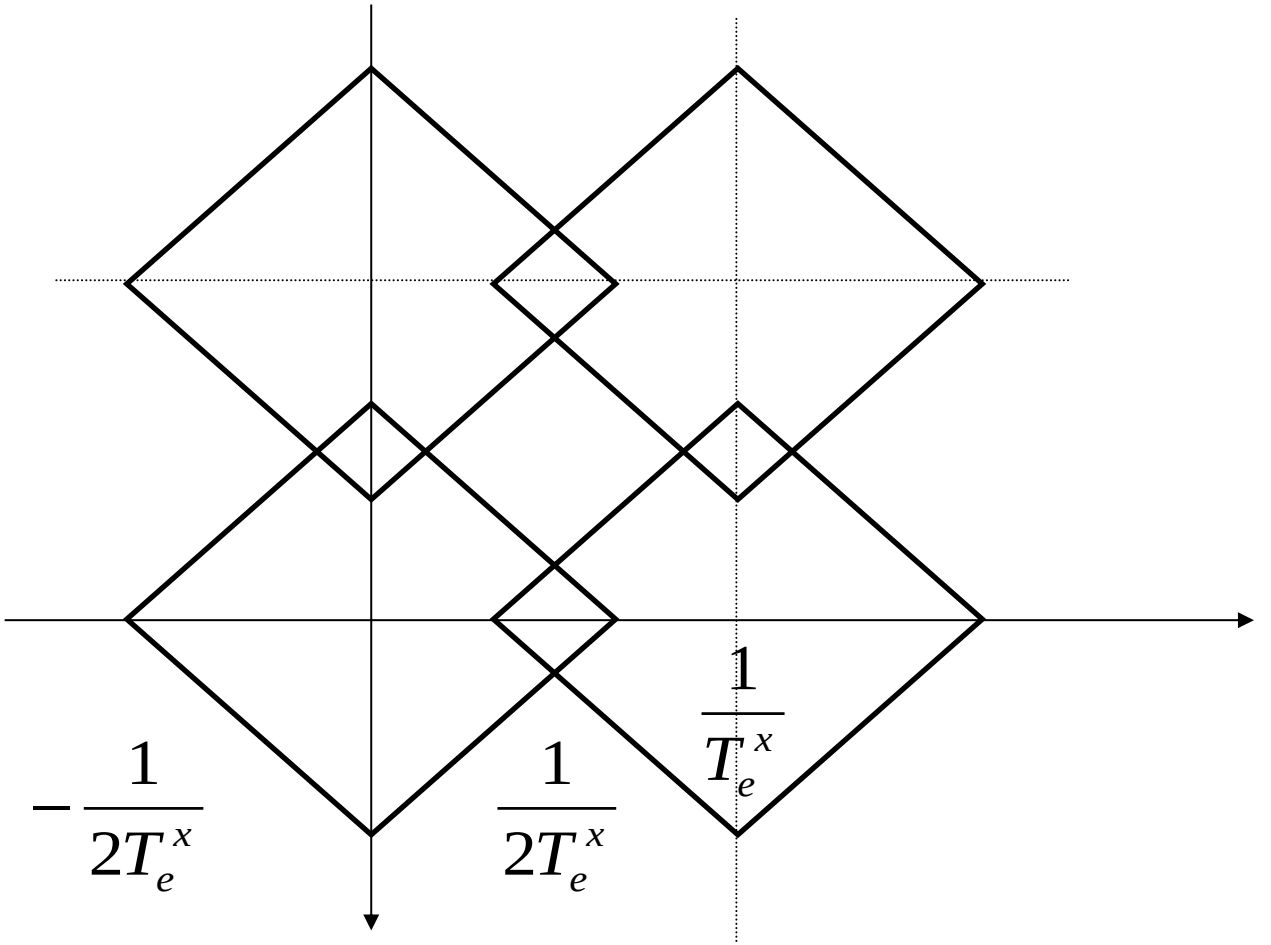
Signal 2d à bande passante limitée



périodisation du spectre

$|F_n(u)|$

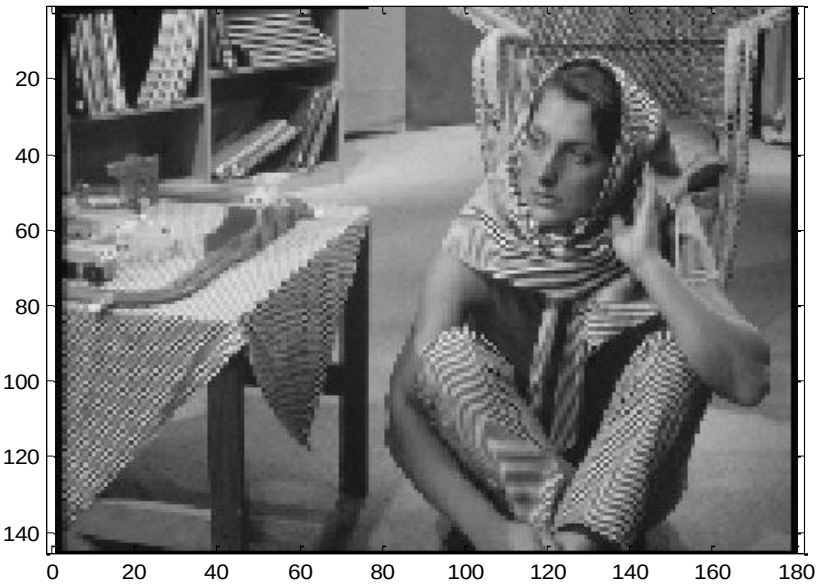
Recouvrement fréquentiel

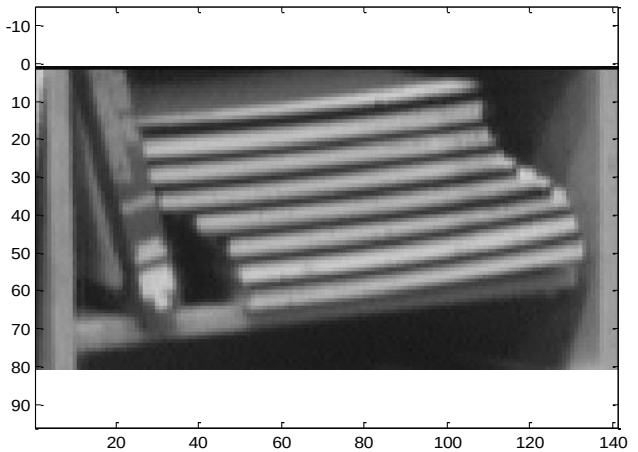


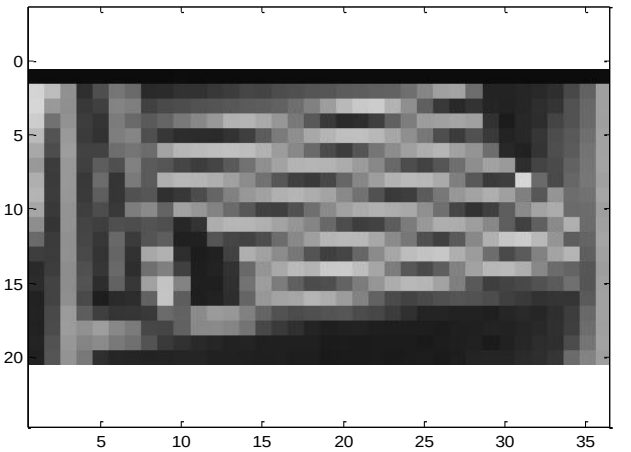
```
clear all
close all

A=imread('barbara.bmp');
figure
image(A)
colormap(gray(256))
axis('equal')
figure
image(A(1:80,140:280))
colormap(gray(256))
axis('equal')
```

```
[Ny,Nx]=size(A);

B=A(1:4:Ny,1:4:Nx);
figure
image(B)
colormap(gray(256))
axis('equal')
figure
image(B(1:20,35:70))
colormap(gray(256))
axis('equal')
```


50
100
150
200
250
300
350
400
450
500
550
0
100
200
300
400
500
600
700


20
40
60
80
100
120
140
0
20
40
60
80
100
120
140
160
180


-10
0
10
20
30
40
50
60
70
80
90
20
40
60
80
100
120
140


0
5
10
15
20
5
10
15
20
25
30
35

Transformée de Fourier

Transformée directe

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy = TF\big(f(x,y)\big)$$

variables fréquentielles

variables spatiales

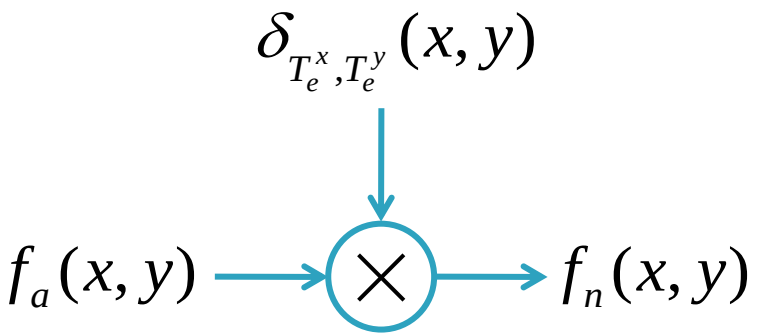
Extension au cas des signaux bidimensionnels (images)

Transformée inverse

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{j2\pi(ux+vy)}dudv = TF^{-1}\big(F(u,v)\big)$$

Echantillonnage régulier 2d

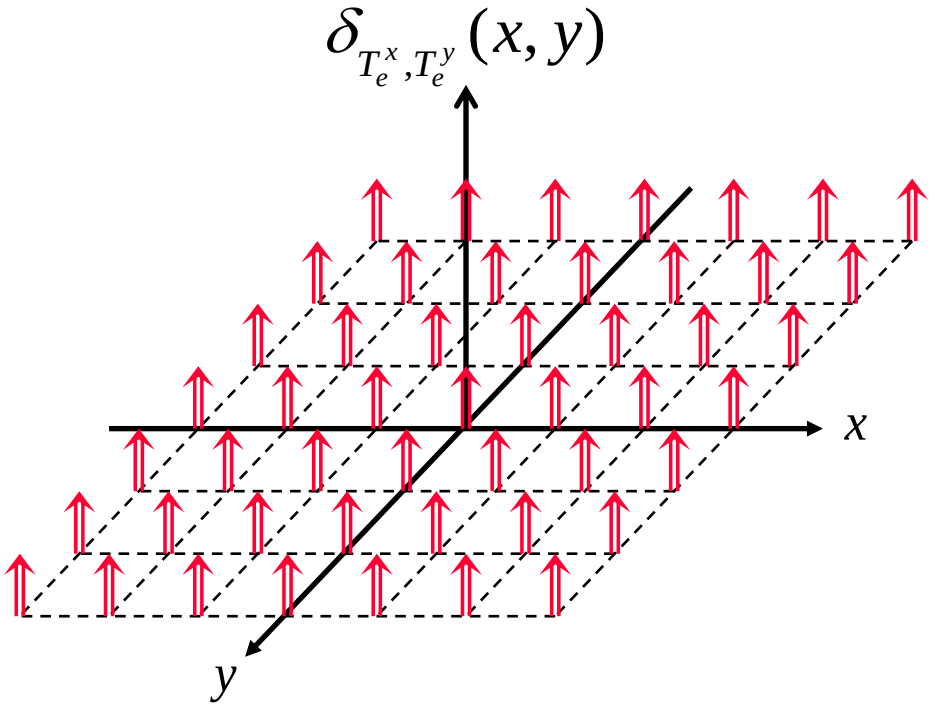
$$f_n(x,y) = \begin{cases} f_a(x,y) & \text{pour} \quad \begin{cases} x = mT_e^x \\ y = nT_e^y \end{cases} \\ 0 & \text{sinon} \end{cases}$$



$$f_n(x,y) = f_a(x,y)\,\delta_{T_e^x,T_e^y}(x,y)$$

séquence discrète

notée $f(mT_e^x, nT_e^y)$ ou $f(m,n)$



Peigne de Dirac 2d

$$\delta_{T_e^x,T_e^y}(x,y) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(x - mT_e^x, y - nT_e^y)$$

Transformée de Fourier discrète 2d

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy = TF\big(f(x,y)\big)$$

Transformée (continue) de Fourier (TF)

échantillonnage spatial

$$F_n(u,v) = \sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty} f(mT_e^x, nT_e^y)e^{-j2\pi\left(umT_e^x+vnT_e^y\right)}$$

Transformée (continue) de Fourier d'une séquence discrète (TFCD)

transformée $\left(f_e^x, f_e^y\right)$-périodique

périodes d'échantillonnage

fenêtrage et échantillonnage des fréquences

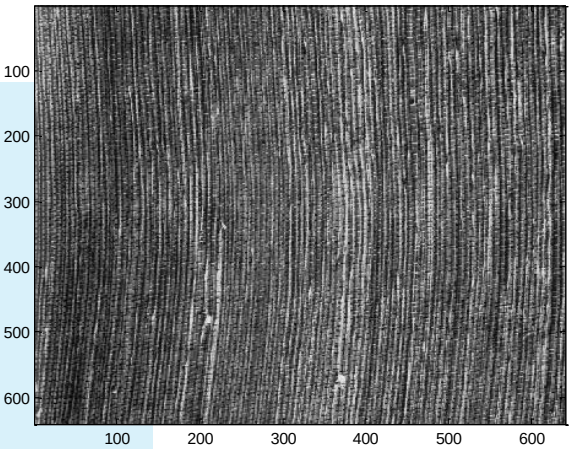
Transformée de Fourier discrète (d'une séquence discrète) (TFD)

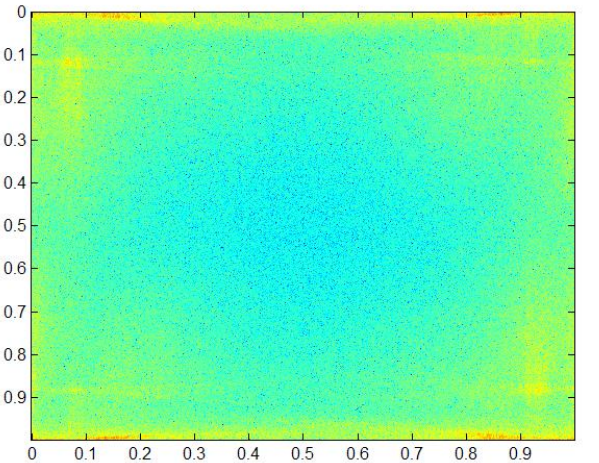
$$F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi\left(k\frac{m}{M}+l\frac{n}{N}\right)}$$

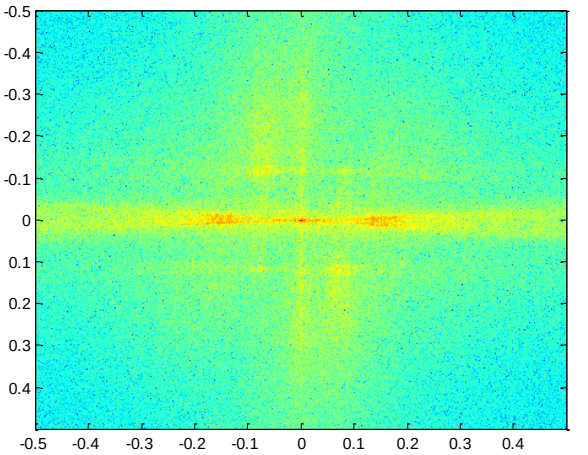
variables discrètes $\begin{cases} k \in [0, M-1] \\ l \in [0, N-1] \end{cases}$

TFD

```
clear all
close all

A=imread('trame.bmp');
figure, image(A)
colormap(gray(256))
[h,w]=size(A);
B=log10(abs(fft2(A)));
fx=linspace(0,1-1/w,w);
fy=linspace(0,1-1/h,h);
figure, imagesc(fx,fy,B);
% w et h pairs
fx=linspace(-0.5,0.5-1/w,w);
fy=linspace(-0.5,0.5-1/h,h);
figure
imagesc(fx,fy,fftshift(B));
```

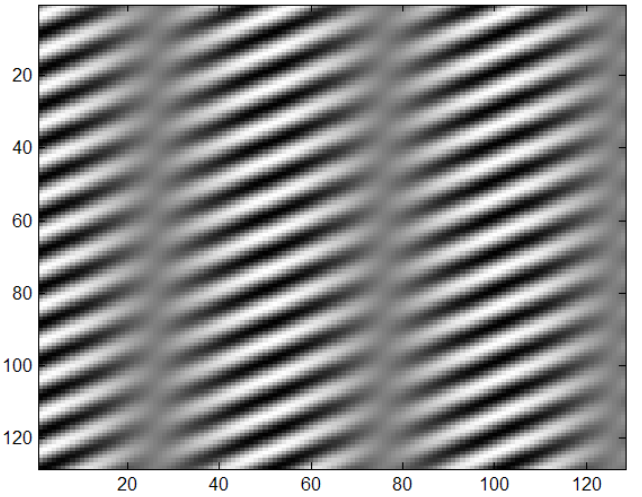


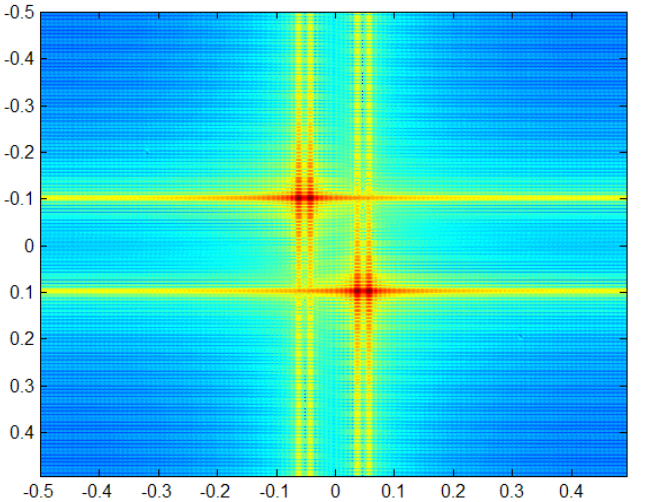




TFD et fenêtrage

```
clear all
close all

N=128;
t=0:N-1;
[X,Y]=meshgrid(t);
I=50*sin(2*pi*0.04*X+2*pi*0.1*Y)+...
  50*sin(2*pi*0.06*X+2*pi*0.1*Y)+...
  0.01*sin(2*pi*0.32*X+2*pi*0.2*Y);
figure, imagesc(I), colormap(gray(256))
If=fftshift(log10(abs(fft2(I,512,512))));
f=linspace(-0.5,0.5-1/N,N);
figure, imagesc(f,f,If)
```



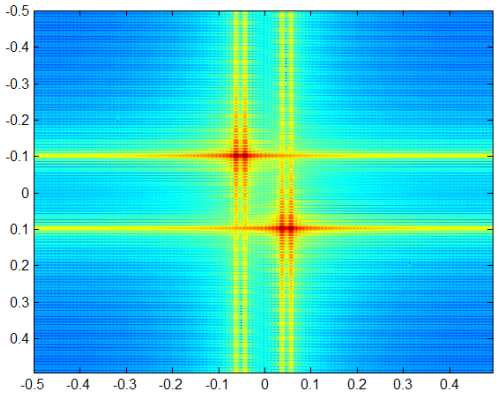


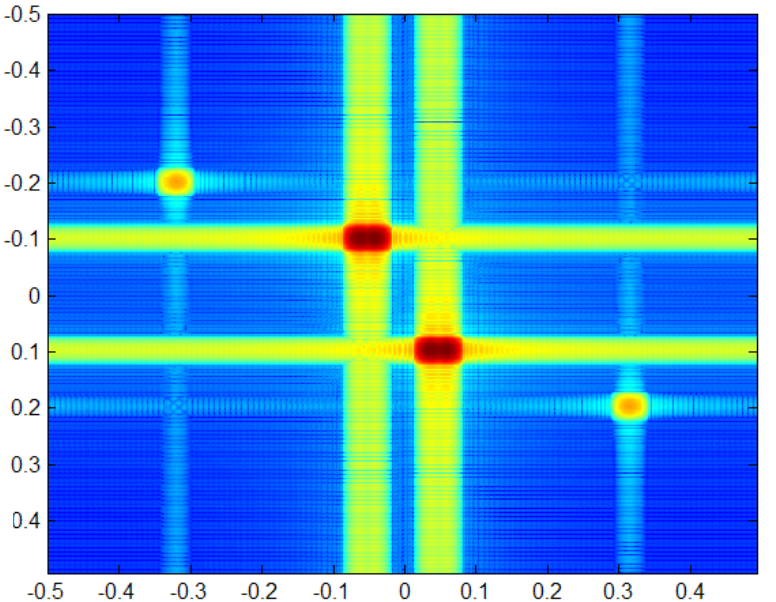
Kaiser séparable

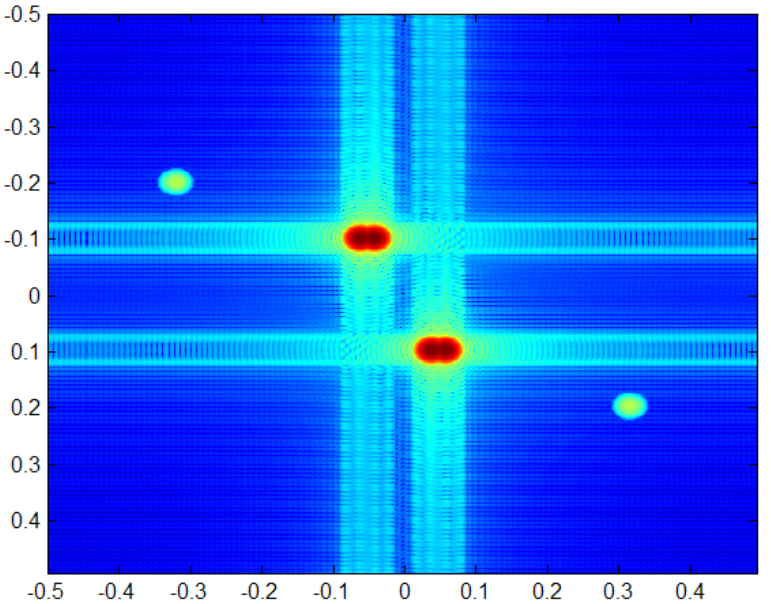
```
W=kaiser(128,10)*kaiser(128,10)';
Iw=I.*W;
If2=fftshift(log10(abs(fft2(Iw,512,512))));
figure, imagesc(f,f,If2)
```

```
M=-64:63;
[X,Y]=meshgrid(M);
R=(X.^2+Y.^2);
Rf=64^2;
a=10;
Wc=besseli(0,a*sqrt(1-R/Rf))/besseli(0,a);
Iw2=I.*Wc;
If3=fftshift(log10(abs(fft2(Iw2,512,512))));
figure, imagesc(f,f,If3)
```

Kaiser à symétrie circulaire


-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4


-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4


-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4

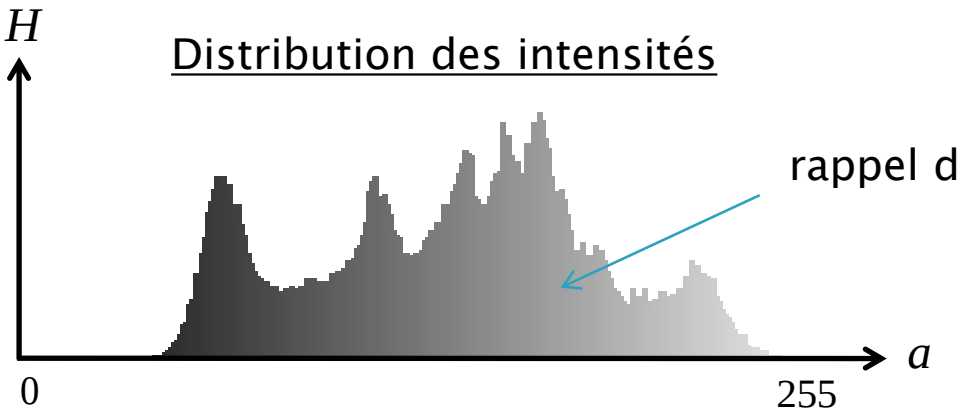
Traitements

- Histogramme
- Filtrage
- Détection de contours

Histogramme (1 /4)

Intensité scalaire

(nombre de pixels)

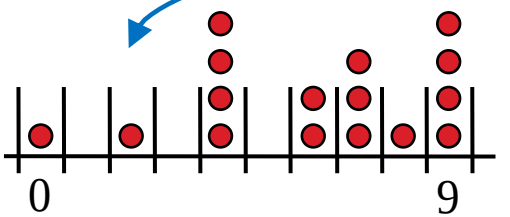


(valeurs d'intensité)

Définition usuelle

$$H(a) = Card(\{ x \in I \,;\, I(x) = a \}) \quad a = 0,\ldots,255$$

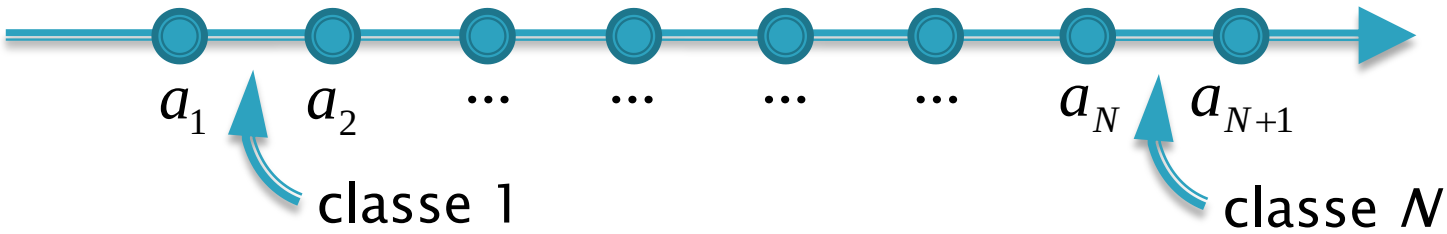
Mode opératoire



Histogramme (2/4)

- Classes (intervalles) d'intensité (index) $a \in [a_1,\ldots,a_{N+1}]$

Définition formelle



- Effectifs (distribution des intensités sur N intervalles ou classes)

$$H(i) = \begin{cases} Card(\{ x \in I ; I(x) < a_{i+1} \}) & i = 1 \\ Card(\{ x \in I ; a_i \le I(x) < a_{i+1} \}) & 1 < i < N \\ Card(\{ x \in I ; a_i \le I(x) \}) & i = N \end{cases}$$

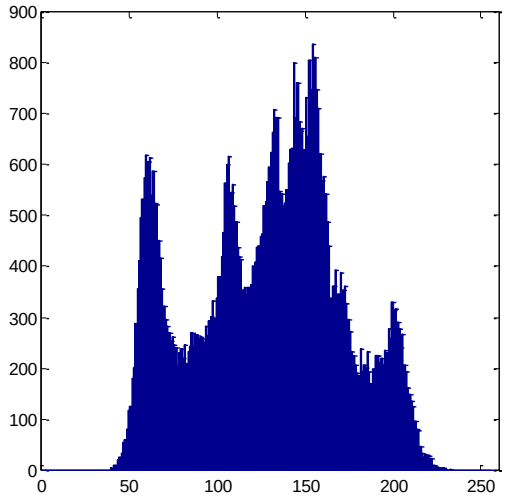
~ densité de probabilité (variables aléatoires continues)

Histogramme (3/4)

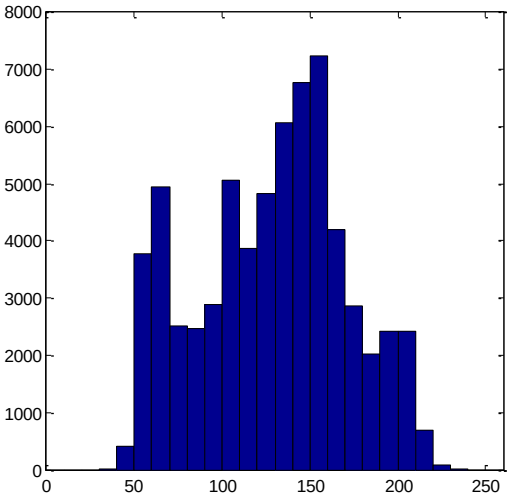
Impact du nombre de classes

amplitude de plus en plus forte

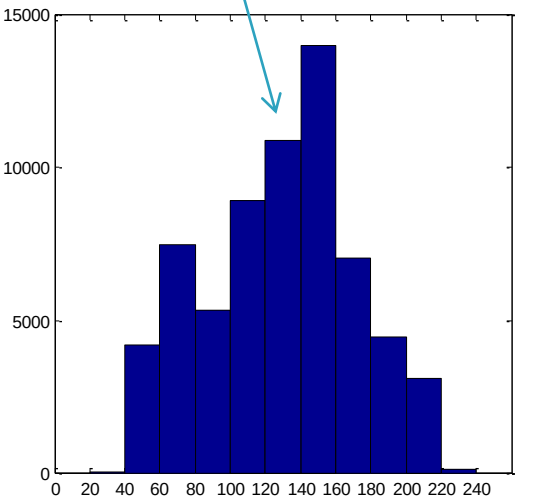
représentativité de plus en plus faible



Histogramme de « 1 en 1 »

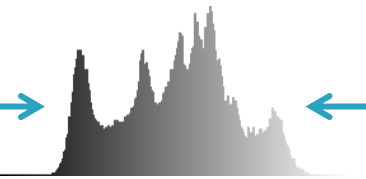


Histogramme de « 10 en 10 »



Histogramme de « 20 en 20 »

Caractérisation statistique



même histogramme

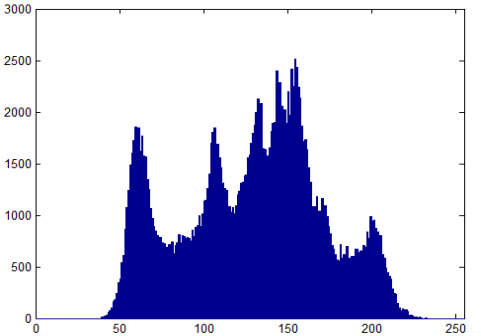
Calcul d'histogramme

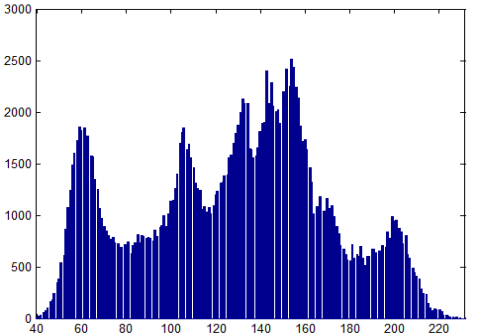
- Affichage d'une image ('lena.bmp')
- Calcul (sur N classes) et affichage de l'histogramme défini par
  - les limites de classes (**histc**)
  - les centres de classes (**hist**)

```
clear all, close all

A=imread('lena.bmp');
figure, imshow(A)

N=256;
v=double(A(:));
a=linspace(0,255,N);
h1=histc(v,a);
figure, bar(a,h1,'histc'), xlim([min(a) max(a)])

[h2,c]=hist(v,N);
figure, bar(c,h2,'hist'), xlim([min(c) max(c)])
```

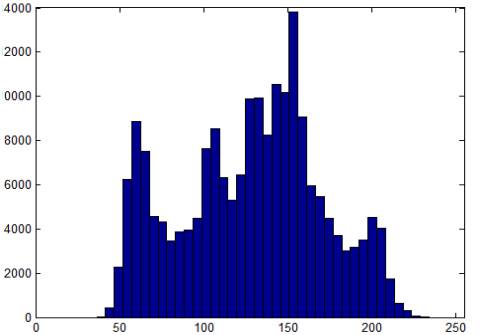
histc **hist**

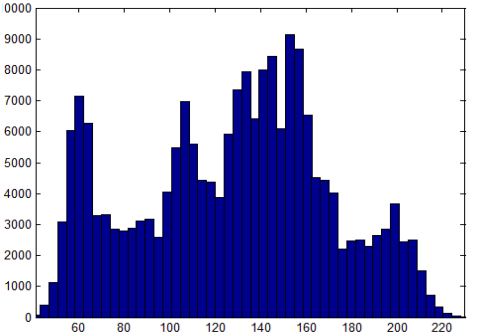
$N = 256$



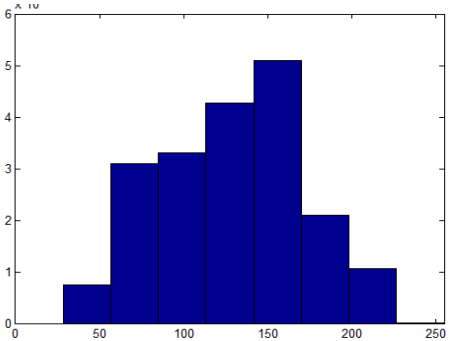


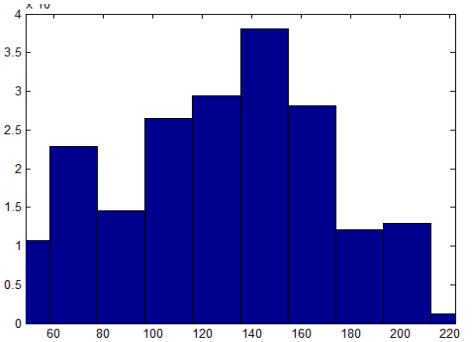
$N = 50$





$N = 10$



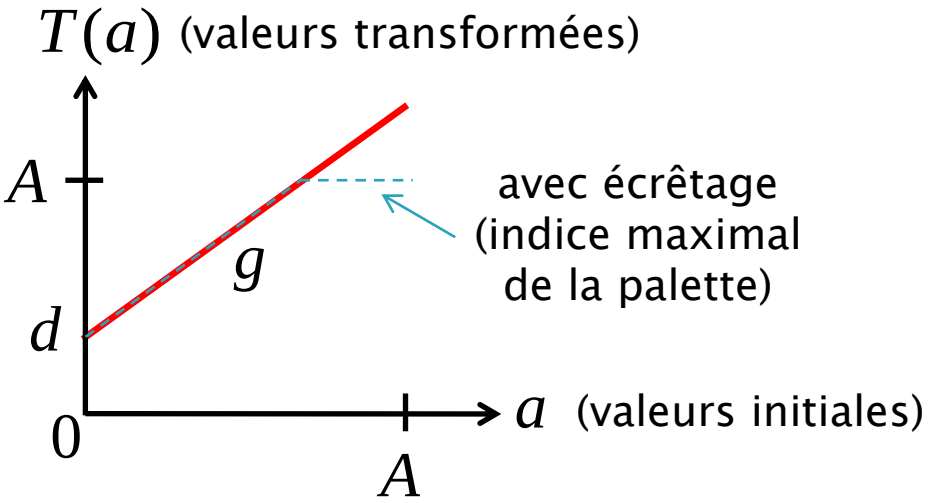


Opérateurs linéaires (1/2)

gain
(augmentation/diminution du contraste)

$$I_S = T(I_e) = gI_e + d$$

décalage
(assombrissement/éclaircissement)



Transformation usuelles

$$I_s = \frac{\left(I_s^{\max} - I_s^{\min}\right)}{\left(I_e^{\max} - I_e^{\min}\right)}\left[I_e - I_e^{\min}\right] + I_s^{\min}$$

Mise à l'échelle

$$I_s = I_e^{\max} + I_e^{\min} - I_e$$

Négatif

Exercice : gain et décalage

- Affichage de l'image 'meb.jpg'
- Multiplication par 2 des intensités
  - Affichage de l'image obtenue $g = 2$
  - Calcul et affichage de l'histogramme
- Division par 2 et décalage de 100 des intensités

  $g = 0.5 , d = 100$
- Complément à 255 des intensités

  $g = -1 , d = 255$

Opérateurs linéaires (2/2)

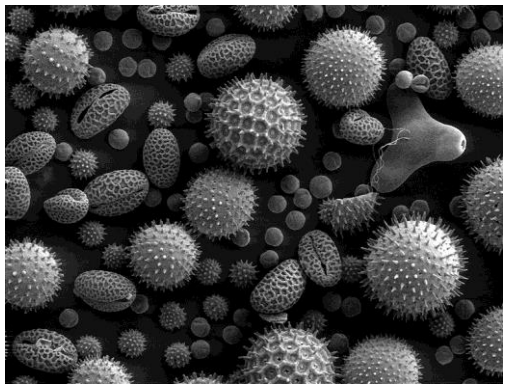
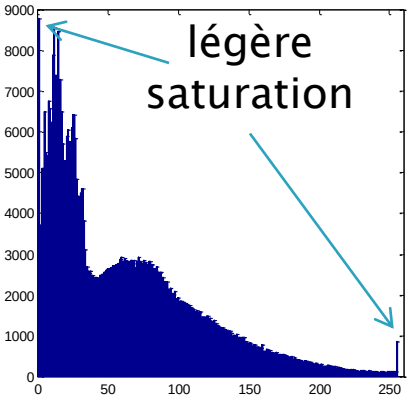
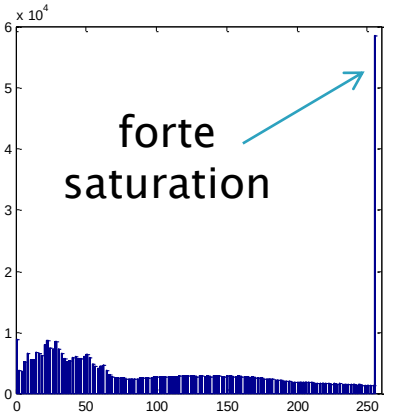
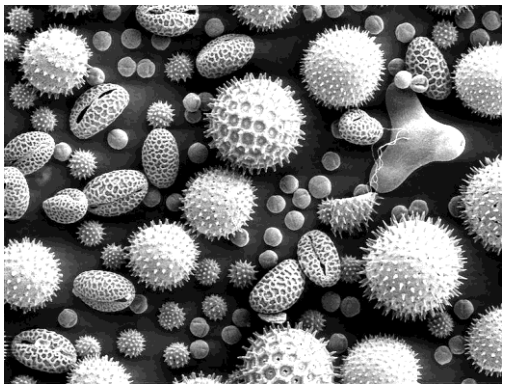


Image initiale

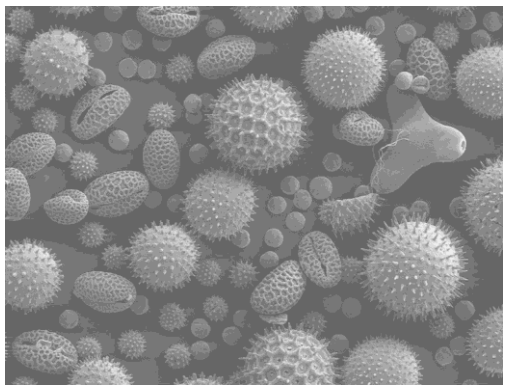




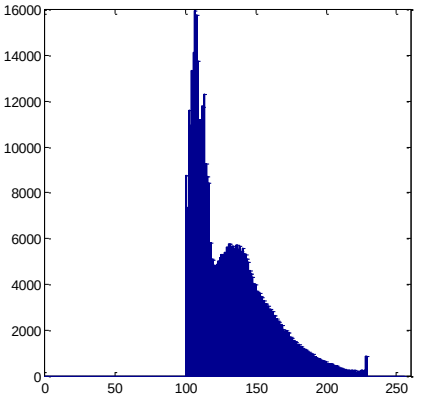
Augmentation de contraste
(avec saturation)



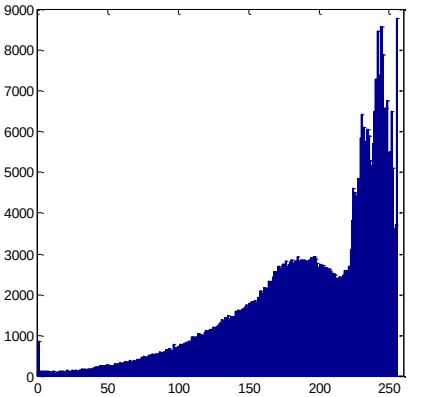
$g = 2$



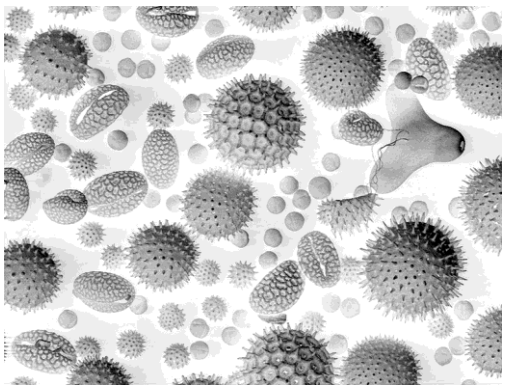
$g = 0.5\,,\, d = 100$



Diminution de contraste
et éclaircissement



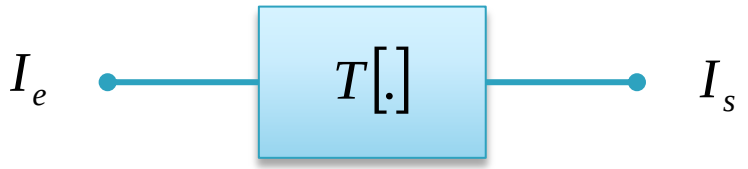
Négatif



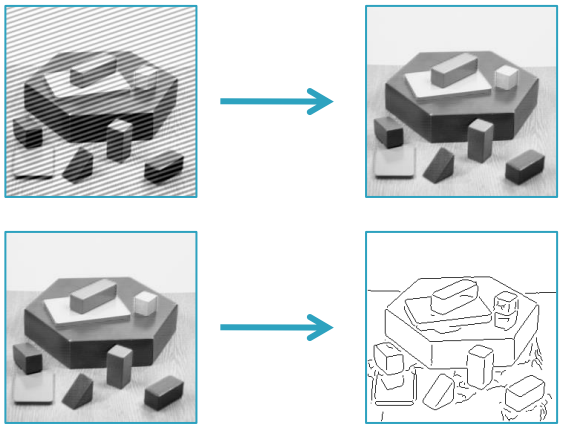
$g = -1\,,\, d = 255$

Filtrage linéaire

Système linéaire



entrée et sortie reliées par un produit de convolution



débruitage

contours

caractérisation par la réponse impulsionnelle $h(t) = T[\delta(t)]$

impulsion

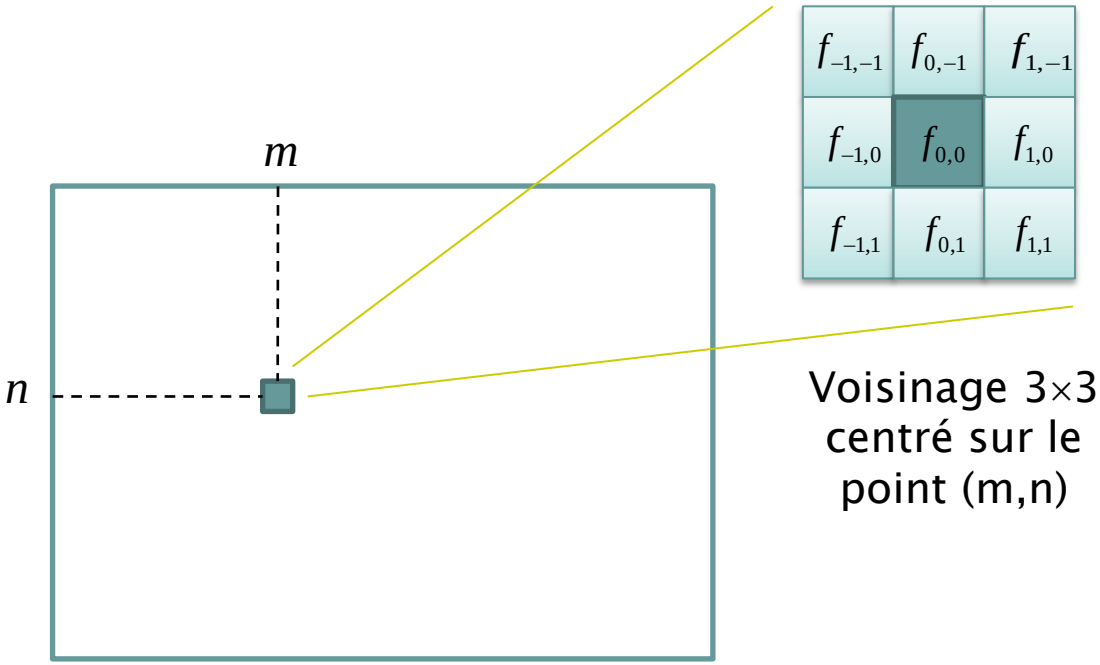
Filtre linéaire RIF

Application d'un filtre linéaire discret à réponse impulsionnelle finie

défini sur $2K+1$ échantillons

$$g(n) = \sum_{k=-K}^{K} f(n-k)h(k) \quad \Leftrightarrow \quad I_s = I_e * H$$

Convolution discrète 2d



$f_{-1,-1}$	$f_{0,-1}$	$f_{1,-1}$
$f_{-1,0}$	$f_{0,0}$	$f_{1,0}$
$f_{-1,1}$	$f_{0,1}$	$f_{1,1}$

Voisinage 3×3 centré sur le point (m,n)

$*$

$h_{-1,-1}$	$h_{0,-1}$	$h_{1,-1}$
$h_{-1,0}$	$h_{0,0}$	$h_{1,0}$
$h_{-1,1}$	$h_{0,1}$	$h_{1,1}$

filtre 3×3

symétrie centrale

$h_{1,1}$	$h_{0,1}$	$h_{-1,1}$
$h_{1,0}$	$h_{0,0}$	$h_{-1,0}$
$h_{1,-1}$	$h_{0,-1}$	$h_{-1,-1}$

$$g(m,n) = \sum_{k=-K}^{K}\sum_{l=-L}^{L} f(m-k,n-l)h(k,l)$$

$$= \sum_{k=-K}^{K}\sum_{l=-L}^{L} f(m+k,n+l)h(-k,-l)$$

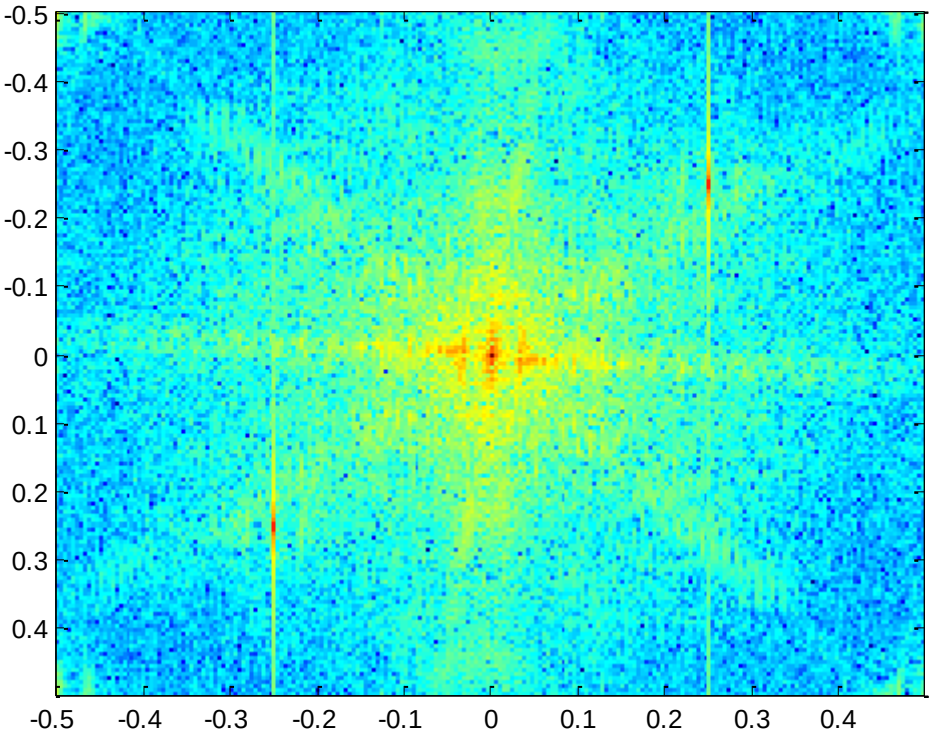
Exemple : convolution 3x3

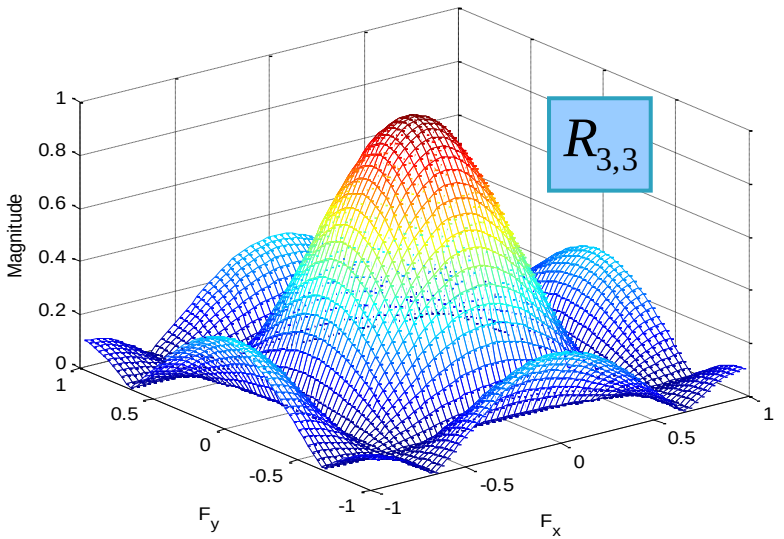
```
clear all
close all

I=[3 1 1 1 1 1;...
   1 1 1 1 1 1;...
   1 1 1 1 1 1;...
   1 1 1 1 1 1]
H=ones(3)/9
Ic1=conv2(I,H)
Ic2=conv2(I,H,'same')
```

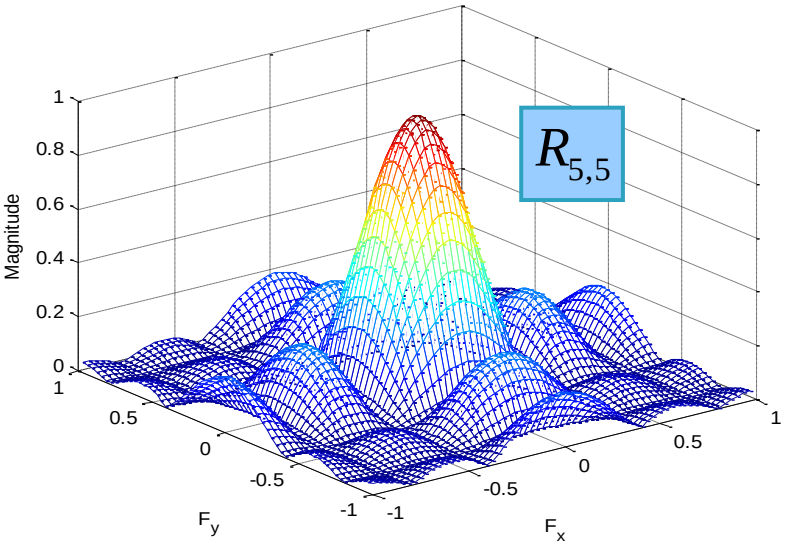
```
I =

     3     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1

H =

    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111

Ic1 =

    0.3333    0.4444    0.5556    0.3333    0.3333    0.3333    0.2222    0.1111
    0.4444    0.6667    0.8889    0.6667    0.6667    0.6667    0.4444    0.2222
    0.5556    0.8889    1.2222    1.0000    1.0000    1.0000    0.6667    0.3333
    0.3333    0.6667    1.0000    1.0000    1.0000    1.0000    0.6667    0.3333
    0.2222    0.4444    0.6667    0.6667    0.6667    0.6667    0.4444    0.2222
    0.1111    0.2222    0.3333    0.3333    0.3333    0.3333    0.2222    0.1111

Ic2 =

    0.6667    0.8889    0.6667    0.6667    0.6667    0.4444
    0.8889    1.2222    1.0000    1.0000    1.0000    0.6667
    0.6667    1.0000    1.0000    1.0000    1.0000    0.6667
    0.4444    0.6667    0.6667    0.6667    0.6667    0.4444
```

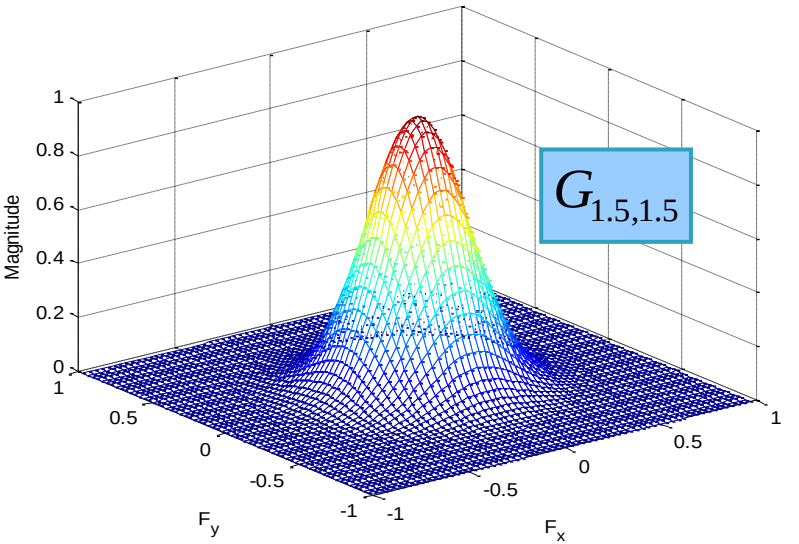
Exercice

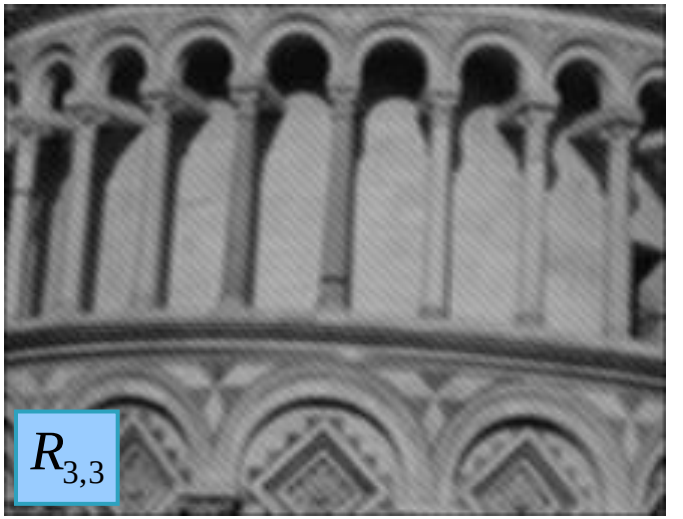
- Affichage de l'image 'pise_ext.bmp'
- Filtrage passe-bas pour éliminer/réduire le bruit
  - Justification du choix (visualisation de la TFD)
  - Vérification de l'adéquation des filtres (freqz2)
- Affichage des images filtrées et de leur spectre
- Calcul et affichage du « bruit » (différence entre l'image initiale et les images filtrées)
- Filtrage coupe-bandes

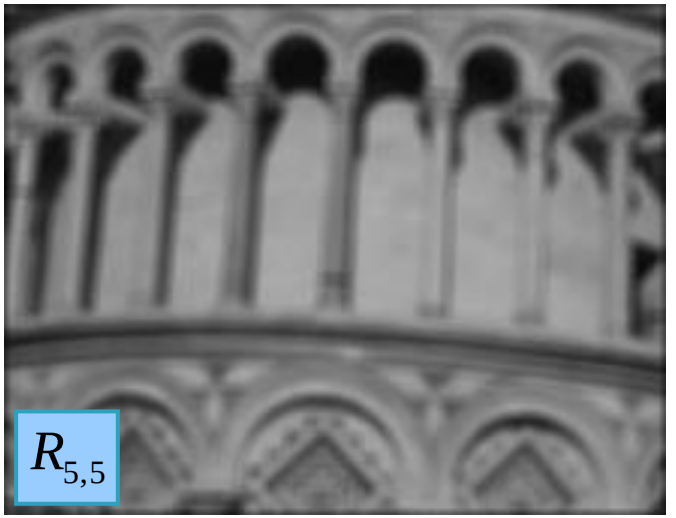

-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4

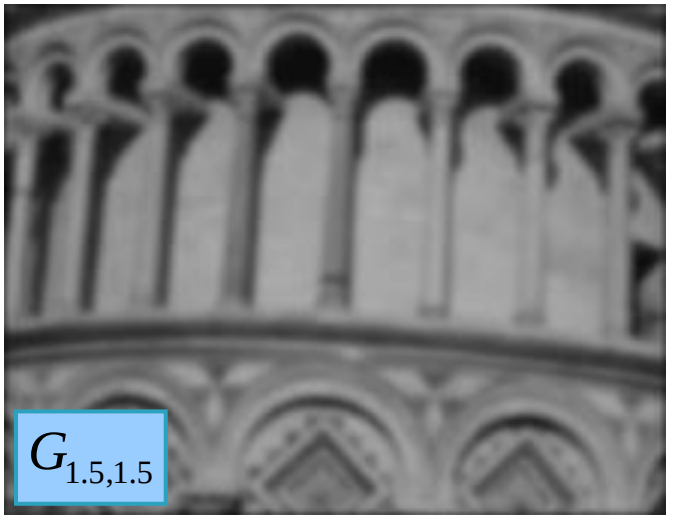

$R_{3,3}$
Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
F_y
-1
-0.5
0
0.5
1
F_x

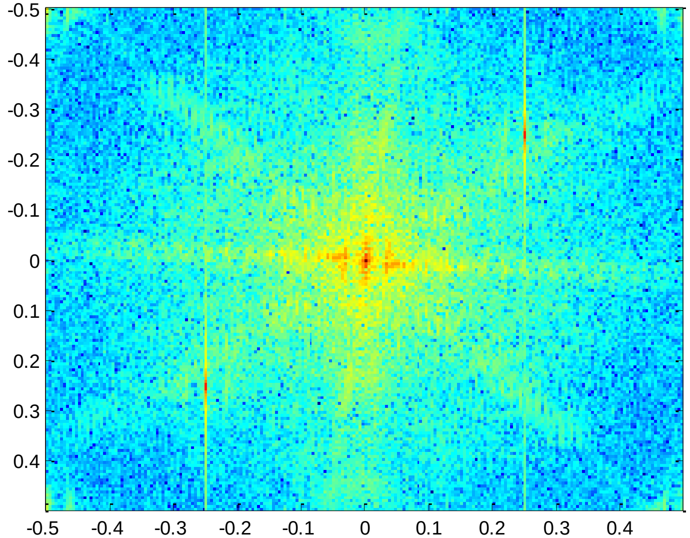

$R_{5,5}$
Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
F_y
-1
-0.5
0
0.5
1
F_x

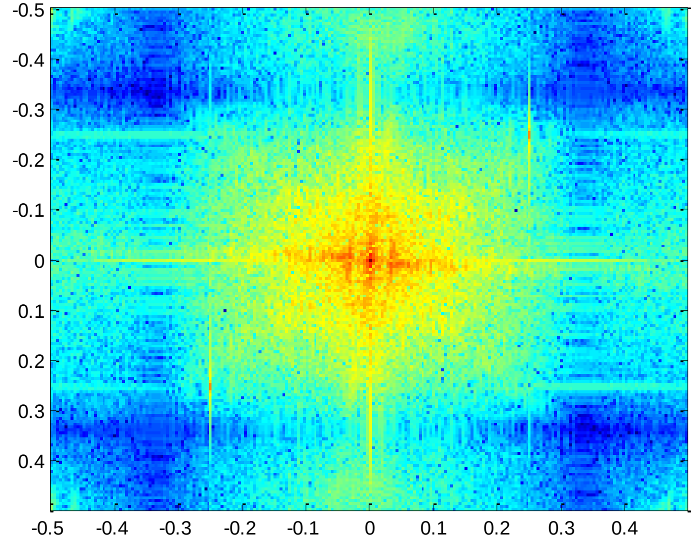

$G_{1.5,1.5}$
Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
F_y
-1
-0.5
0
0.5
1
F_x

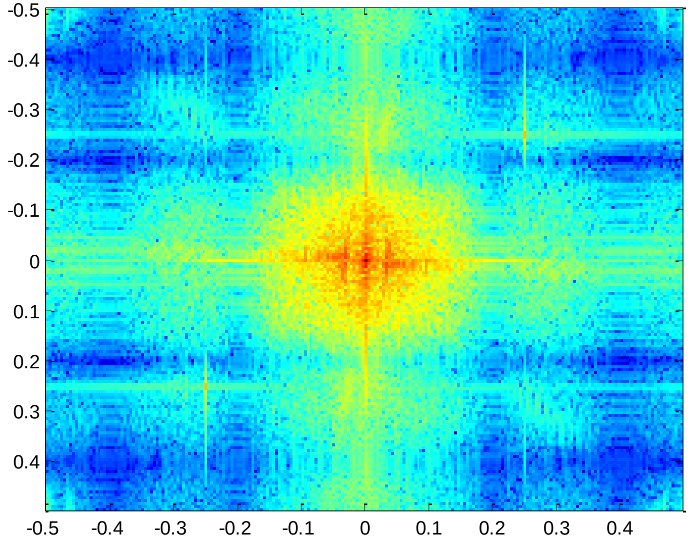

$R_{3,3}$

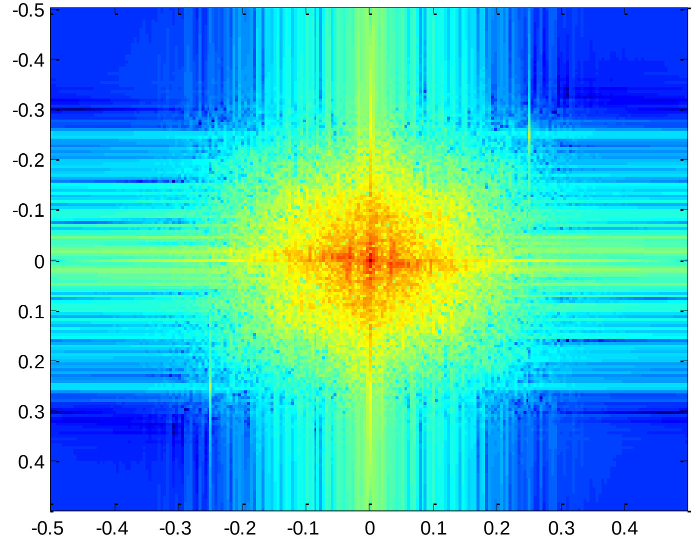

$R_{5,5}$


$G_{1.5,1.5}$

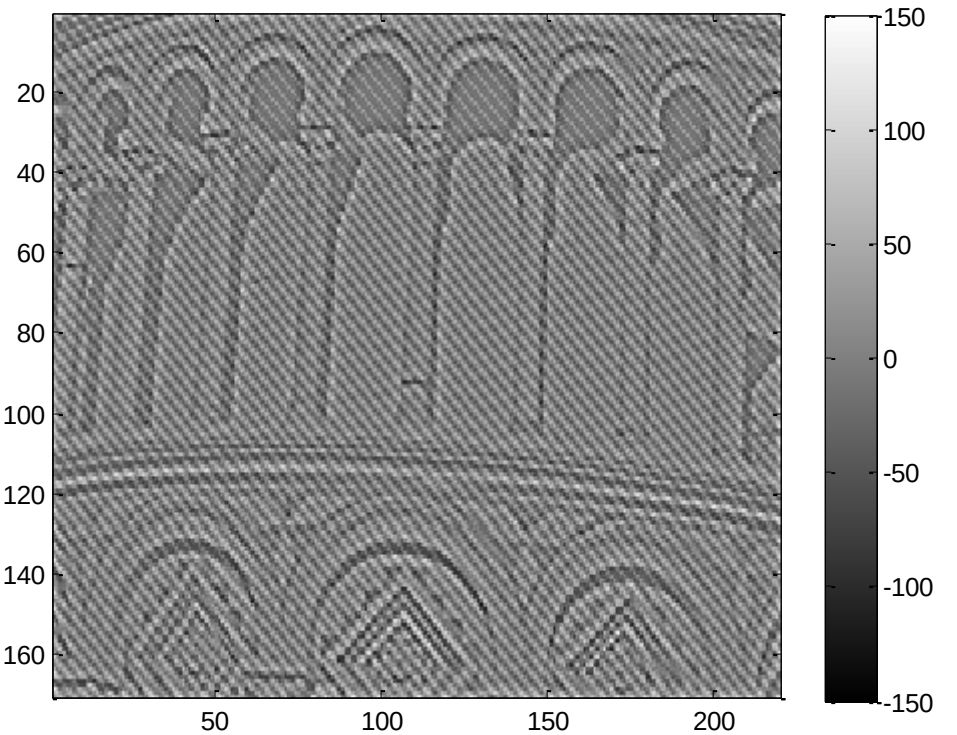

-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4


-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4


-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4

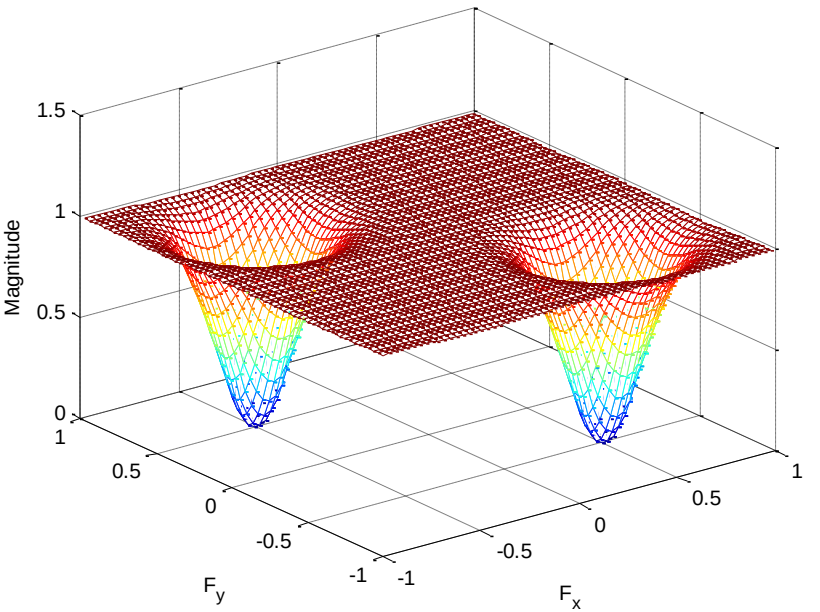

-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4

```
figure, imagesc(A-B,[-150 150]),
colormap(gray(256)), colorbar
figure, imagesc(A-C,[-150 150]),
colormap(gray(256)), colorbar
figure, imagesc(A-D,[-150 150]),
colormap(gray(256)), colorbar
```



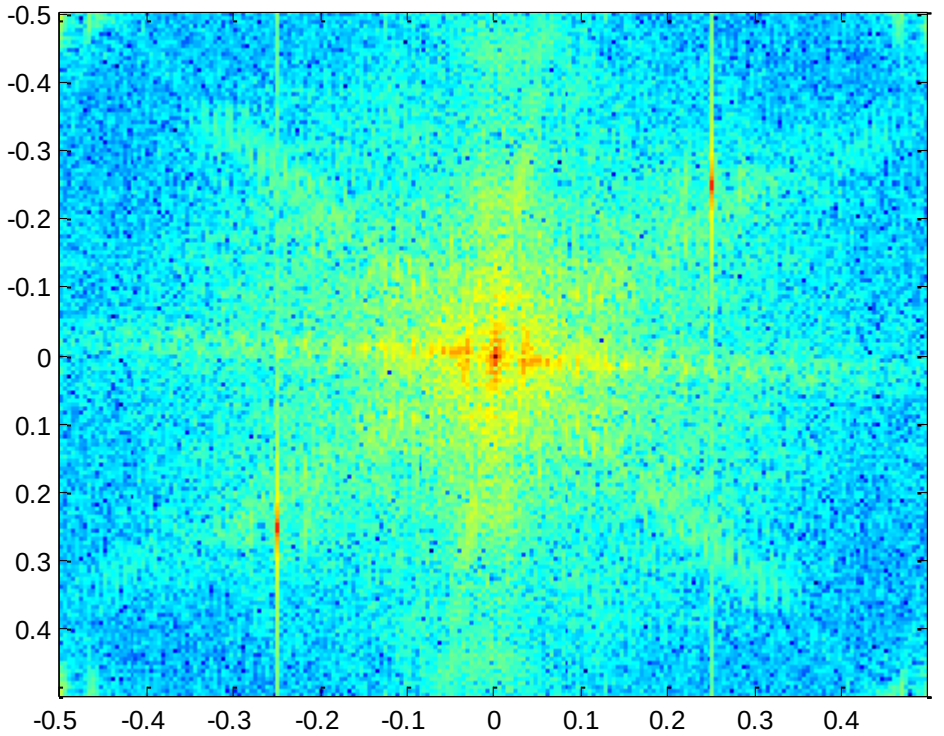
Filtrage coupe-bandes (1/3)

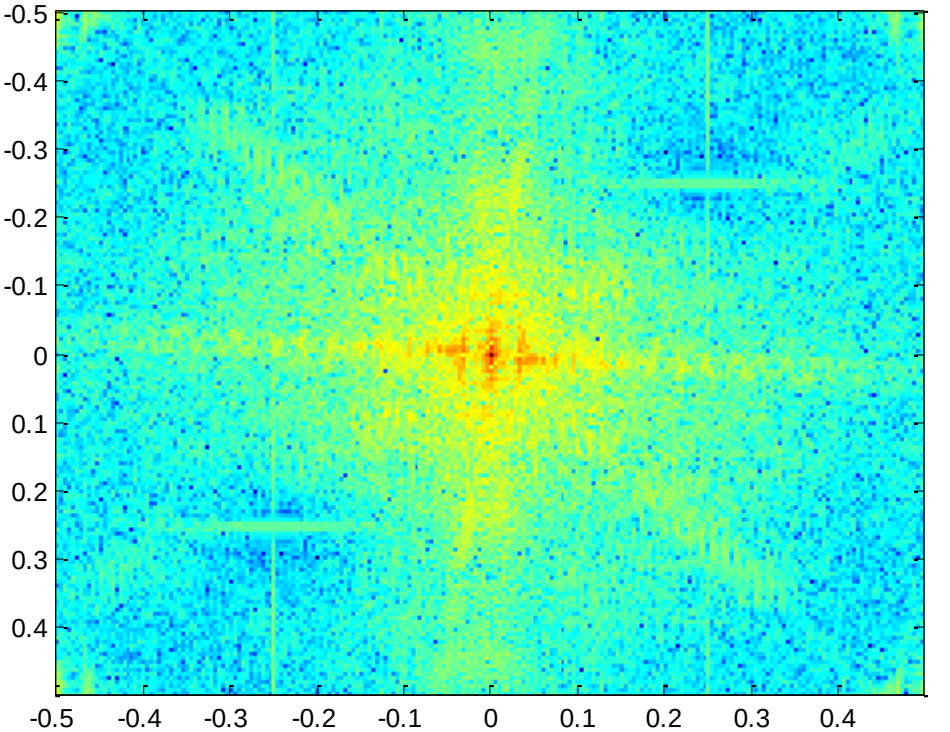
```
sigma=2.0;
fox=0.25;
foy=-0.25;
t=ceil(3*sigma);
[X,Y]=meshgrid(-t:t);
H4=exp(-(X.^2+Y.^2)/(2*sigma^2));
H4=H4./sum(H4(:));
H4=2*cos(2*pi*(X*fox+Y*foy)).*H4;
H4=-H4;
H4(t+1,t+1)=H4(t+1,t+1)+1;
figure, freqz2(H4)
E=conv2(A,H4,'same');
figure, imshow(uint8(E));
IfE=fftshift(log10(abs(fft2(E))));
figure, imagesc(fx,fy,IfE)
```



Filtrage coupe-bandes (2/3)

Filtrage coupe-bandes (3/3)

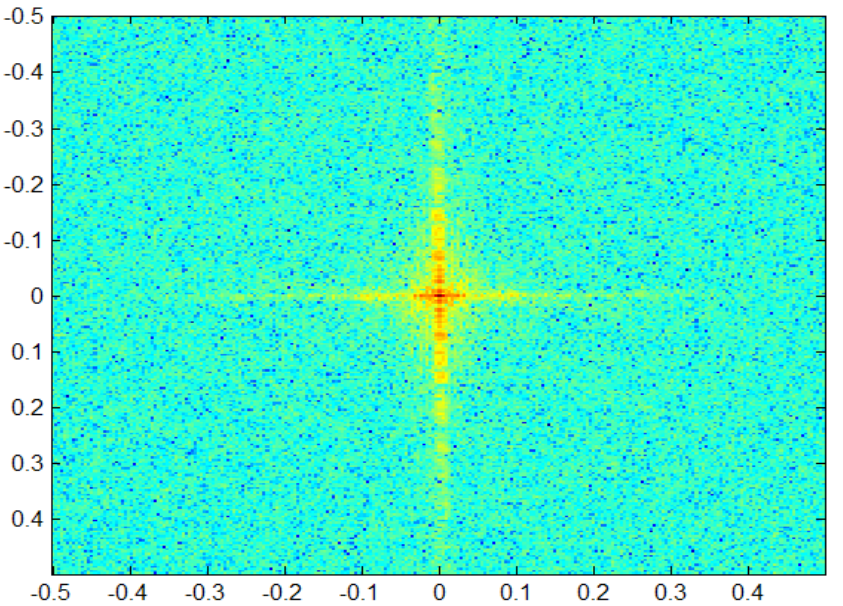


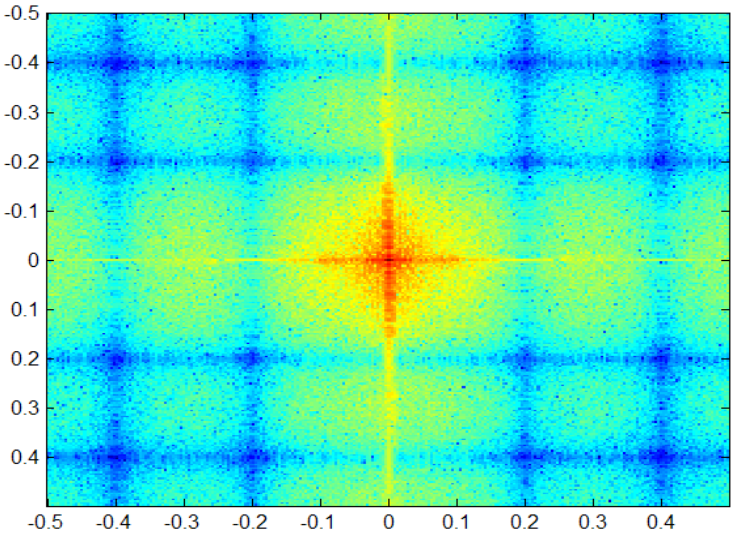


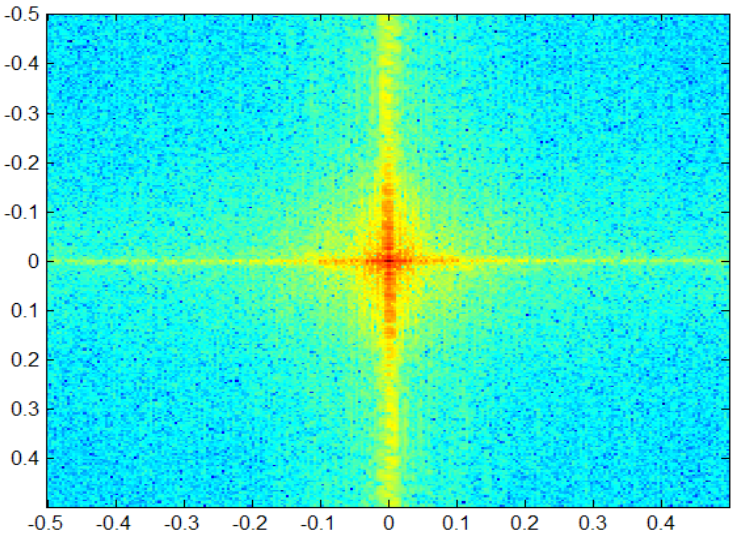
Filtrage d'ordre

- Affichage de l'image 'couloir.tif'
- Application d'un filtre RIF passe-bas
- Comparaison avec un filtre non linéaire (médian)

```
clear all, close all

A=double(imread('couloir.tif'));
figure, imshow(uint8(A))
[h,w]=size(A);
fx=linspace(-0.5,0.5-1/w,w);
fy=linspace(-0.5,0.5-1/h,h);
IfA=fftshift(log10(abs(fft2(A))));
figure, imagesc(fx,fy,IfA)
H1=ones(5)/25;
B=conv2(A,H1,'same');
figure, imshow(uint8(B));
IfB=fftshift(log10(abs(fft2(B))));
figure, imagesc(fx,fy,IfB)
C=medfilt2(A, [5 5]);
figure, imshow(uint8(C));
IfC=fftshift(log10(abs(fft2(C))));
figure, imagesc(fx,fy,IfC)
```


-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4


-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4


-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4

Détection de contours

```
clear all
close all

A=double(imread('cameraman.tif'));
figure, imshow(uint8(A))
[X,Y]=meshgrid(-5:5);
sigma=1.5;
Hx=-X.*exp(-(X.^2+Y.^2)/(2*sigma^2));
Hx=Hx/sum(Hx(:));
Hy=-Y.*exp(-(X.^2+Y.^2)/(2*sigma^2));
Hy=Hy/sum(Hy(:));
Gx=conv2(A,Hx,'same');
Gy=conv2(A,Hy,'same');
G=(Gx.*Gx+Gy.*Gy).^0.5;
Figure, imshow(G, [0 50])
colormap(flipud(gray(256)))
```

$\sigma = 0{,}75$

$\sigma = 1{,}5$

$\sigma = 2,5$

Syntaxes et fonctions utiles

$C = A + B$

$C = A.*B;$

$B = A';$

$B = A > 50$

$B = A(10:end, 5:10:100, :)$

$v = A(:)$

$A = reshape(v, [h\ w])$

$B = double(A)$

size
imread, imwrite
cat
image, imagesc, imshow
colorbar, colormap, axis, xlim, ylim
min, max
fix, floor, round
meshgrid
linspace, fft2, fftshift
hist, histc, bar
ginput
conv2

ENSC - 2ème année

Initiation au Traitement des Images