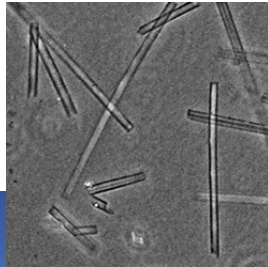
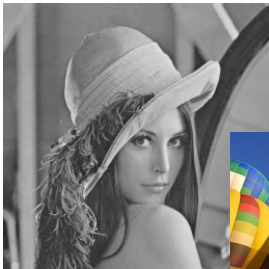


Initiation au Traitement des Images



Organisation

- ▶ Cours (8h, Y. Berthoumieu)
- ▶ Enseignement intégré (8h, M. Donias)
 - Cours et mise en oeuvre dans l'environnement Matlab
- ▶ Travaux Pratiques (8h, M. Donias)
 - Chroma-Keying, Détramage, Contours/Rehaussement
- ▶ Evaluations
 - QCM de cours
 - Examen écrit sans documents
 - Présentiel
 - Rapport (par binôme)

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



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

Matrice R

« Vraies » couleurs

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

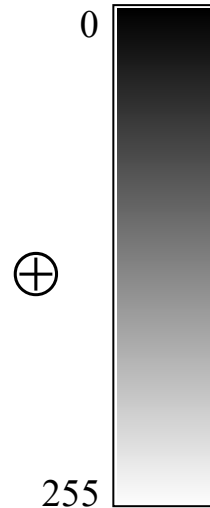
Format usuel : entiers 0 à 255

Définition (2 / 2)



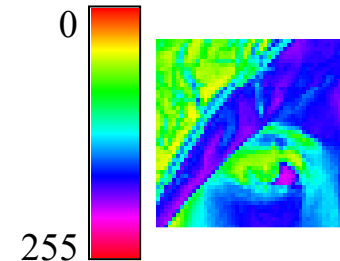
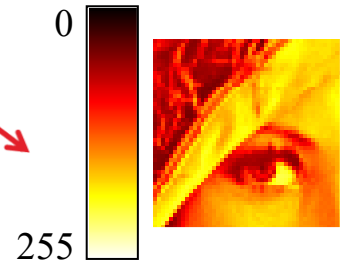
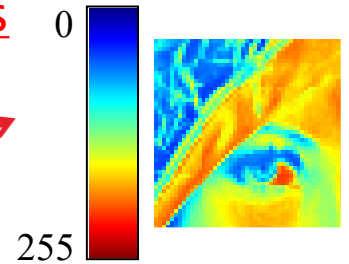
60	64	62	62	55	51
59	73	98	90	66	54
72	105	167	170	120	74
88	91	188	202	184	150
103	77	191	205	203	190
75	131	208	209	207	202

Intensité scalaire



Palette
(niveaux de gris)

Autres palettes



Couleurs indexées ou « fausses » couleurs

- Scalaire (index de table, conversion souvent nécessaire)
- Palette (table de couleurs)

Plan

- ▶ Images couleur
 - Lecture
 - Visualisation
 - Synthèse
- ▶ Espaces caractéristiques
 - Couleur/Amplitude
 - Spatial
 - Fréquentiel
- ▶ Traitements
 - Filtrage linéaire
 - Filtrage non-linéaire

Un exemple en « vraies » couleurs

```
>> A=imread('pool.tif');
```

```
>> whos
```

Name	Size	Bytes	Class
A	383x510x3	585990	uint8

```
>> figure, imshow(A)
```

```
>> A(:, :, 1)
```

```
>> A(:, :, 2)
```

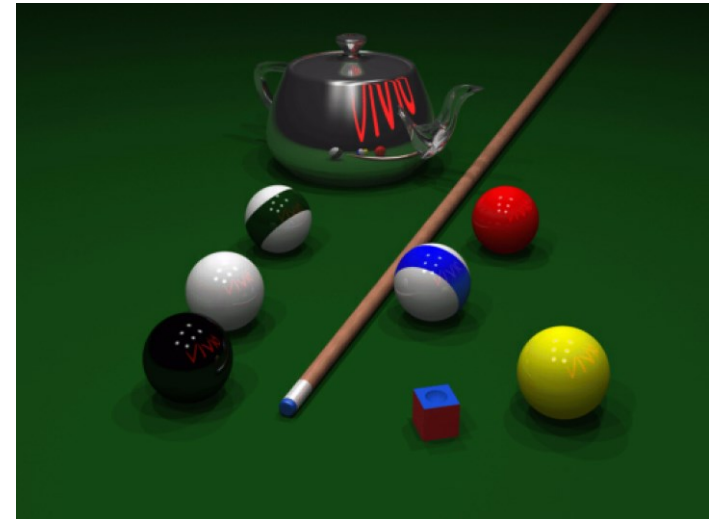
```
>> A(:, :, 3)
```

format numérique

hauteur

largeur

« vraies »
couleurs



Un exemple en couleurs indexées

```
>> A=imread('lena.bmp');  
>> whos
```

Name	Size	Bytes	Class
A	512x512	262144	uint8

```
>> figure, imshow(A), colorbar  
>> figure, imshow(A, Colormap=gray(8)), colorbar
```



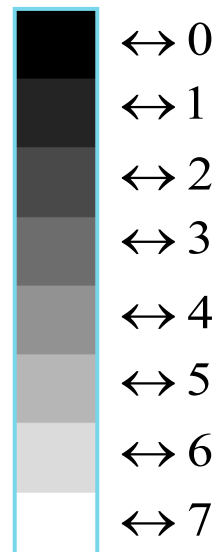
Palette ?

```
>> gray(8)
```

```
ans =
```

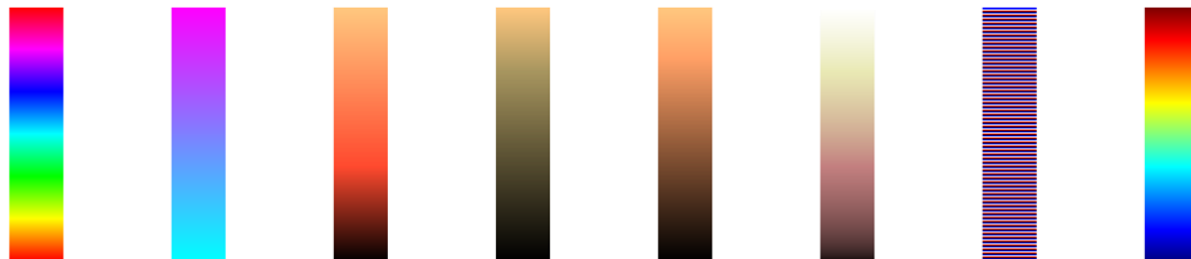
0	0	0
0.1429	0.1429	0.1429
0.2857	0.2857	0.2857
0.4286	0.4286	0.4286
0.5714	0.5714	0.5714
0.7143	0.7143	0.7143
0.8571	0.8571	0.8571
1.0000	1.0000	1.0000

R, G, B



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

256 niveaux par défaut



Fonctions Matlab de visualisation

	image	imagesc	imshow
Formats	uint8, uint16, double		uint8, uint16, single, double, ...
Palette initiale	parula (256 niveaux)		gray (256 niveaux)
Intervalles (couleurs indexées)	[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 [0,65535] uint16 [0,1] single, double
Grossissement initial	variable		1
Ratio L/H initial	variable		1 (pixels carrés)
Divers		Mise à l'échelle des intensités	Mise à l'échelle des intensités

Comportement dépendant de la version (20XXa/b) et du système d'exploitation (Windows/Linux/MaxOS)

Toolbox « Image Processing »

Méthodologie d'affichage (1 / 4)

► Bilan des caractéristiques de l'image

- Dimensions spatiales
- Format numérique
- Intervalles d'intensité
- Codage
 - ✓ « vraies » couleurs
 - ✓ couleurs indexées

```
whos A  
size(A)  
min(A(:))  
max(A(:))
```

► Paramétrage de la fonction d'affichage

- Selon un contexte
 - ✓ de « fidélité »
 - ✓ « informationnel »
 - ✓ de comparaison
- Choix optionnels : palette, ratio L/H, intervalles d'intensité, etc.

Méthodologie d'affichage (2/4)

Contexte « de fidélité »

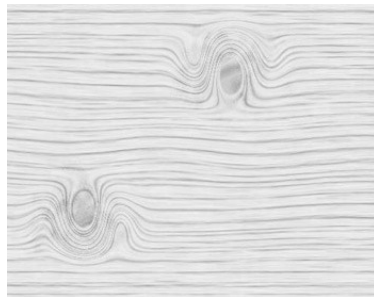
- Respect des intensités initiales
- `imshow(A)`



insaturée



sombre



claire

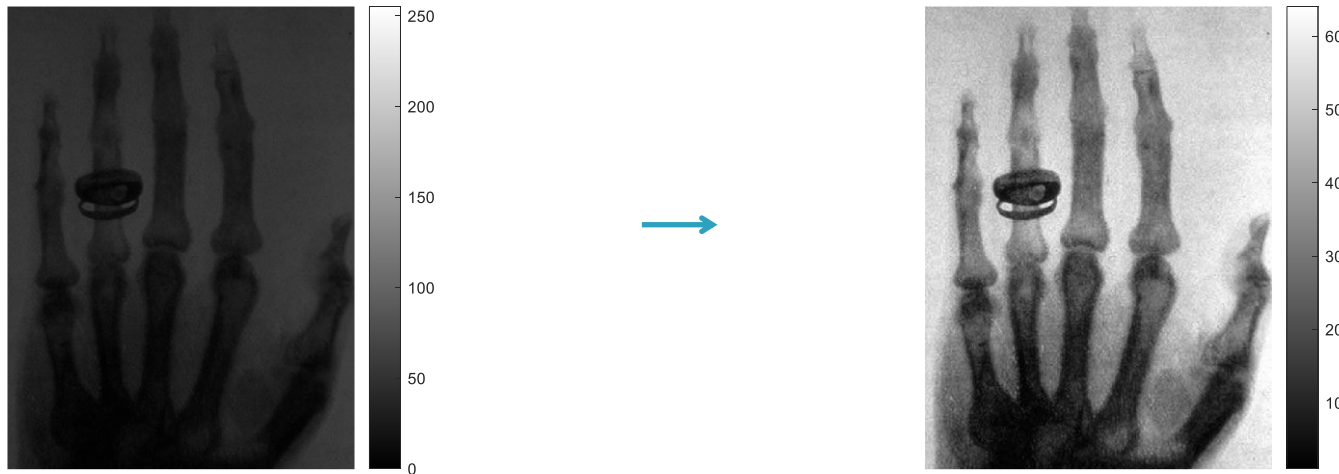


saturée

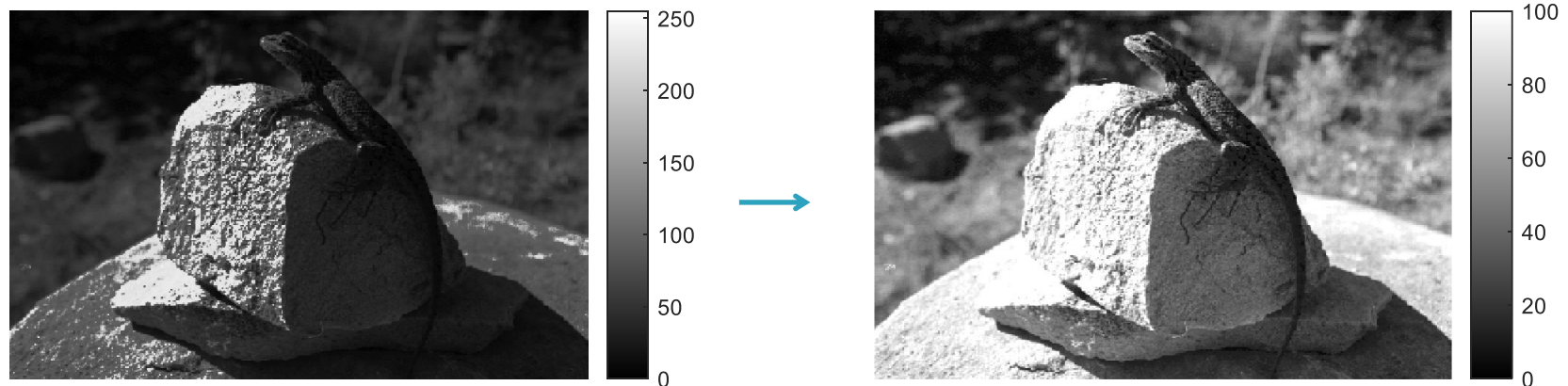
Méthodologie d'affichage (3/4)

Contexte « informationnel » (couleurs indexées uniquement)

- Etalement automatique des intensités (`imshow(A, [ ])`)



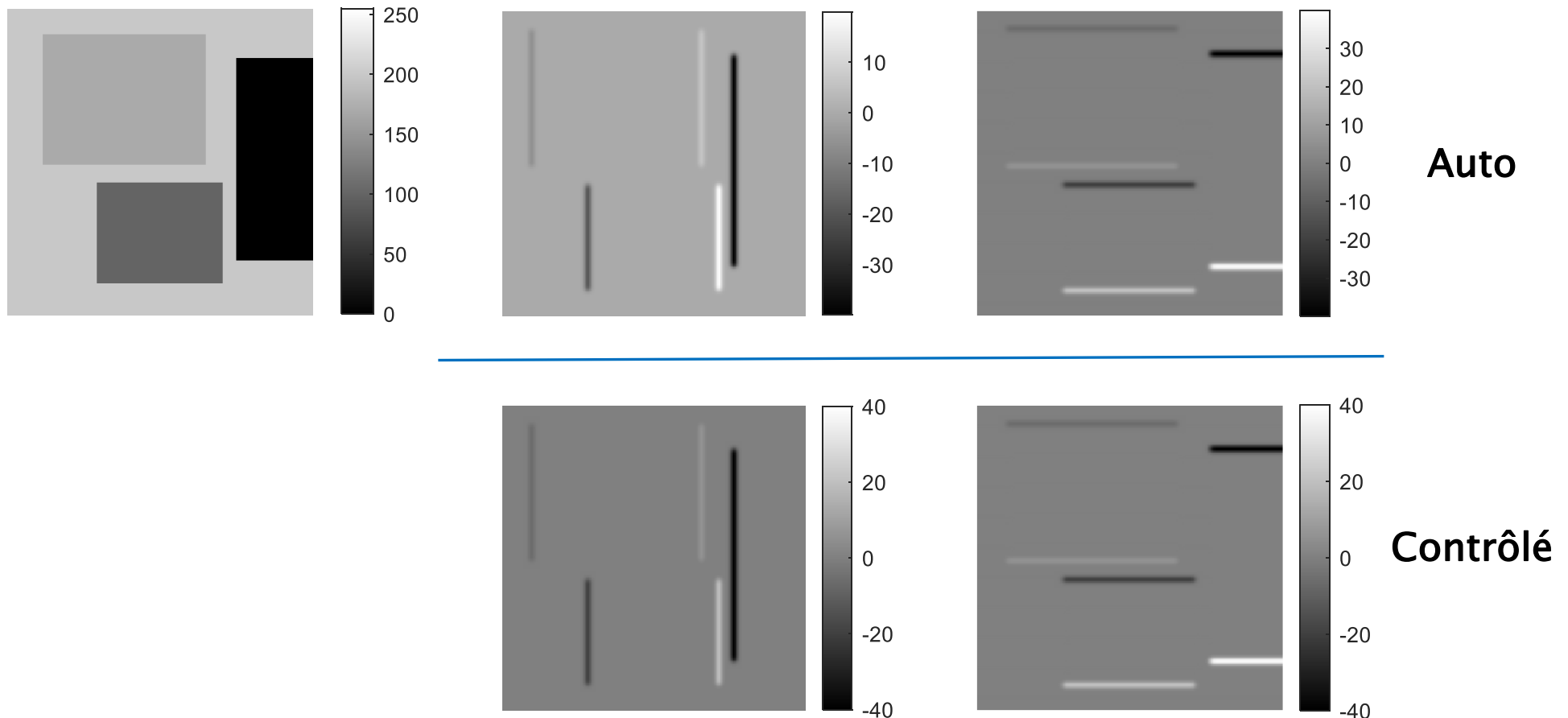
- Etalement contrôlé des intensités (`imshow(A, [a b])`)



Méthodologie d'affichage (4/4)

Contexte de comparaison (couleurs indexées uniquement)

- Etalement contrôlé et identique des intensités (`imshow(...,[a b])`)



Cas d'usage d'affichage

```
clear, close all  
  
s=load('challenge.mat');  
I=s.A; % Challenge 1
```

Objectif : obtenir des affichages
« convenables » répondant à une
problématique

Cas	Donnée à afficher	Problématique
1	I = s.A	Format numérique
2	I = s.B	Intervalle d'intensité
3	I = s.C	Palette continue
4	I = s.D	Palette discrète

Cas 1 : adéquation format/intervalle

```
clear, close all  
  
s=load('challenge.mat');  
I=s.A;
```

- ▶ Déterminer
 - Format numérique
 - Intervalle d'intensité
- ▶ Afficher
 - Fonction **imshow**
- ▶ Adapter
 - Format numérique
 - Intervalle d'intensité

Cas 2 : intervalle d'intensité

```
clear, close all  
  
s=load('challenge.mat');  
I=s.B;
```

- ▶ Déterminer
 - Format numérique
 - Intervalle d'intensité
- ▶ Adapter
 - Intervalle d'intensité le plus « pertinent »

Cas 3 : palette continue

```
clear, close all  
  
s=load('challenge.mat');  
I=s.C;
```

- ▶ Déterminer
 - Format numérique
 - Intervalle d'intensité
- ▶ Identifier
 - Palette « sémantique »

Cas 4 : palette « discrète »

```
clear, close all  
  
s=load('challenge.mat');  
I=s.D;
```

- ▶ Déterminer
 - Format numérique
 - Intervalle d'intensité
- ▶ Afficher
 - Une couleur par continent
 - Des couleurs « choisies »

Synthèse en « vraies couleurs »

```
clear, close all

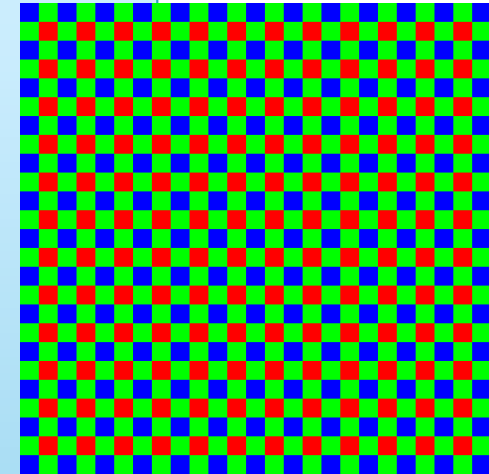
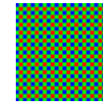
Nx=25;
Ny=25;

[X,Y]=meshgrid(0:Nx-1,0:Ny-1);

R=(1-mod(X+1,2)).*(1-mod(Y+1,2));
G=mod(X+Y,2);
B=(1-mod(X,2)).*(1-mod(Y,2));

I=cat(3,R,G,B);

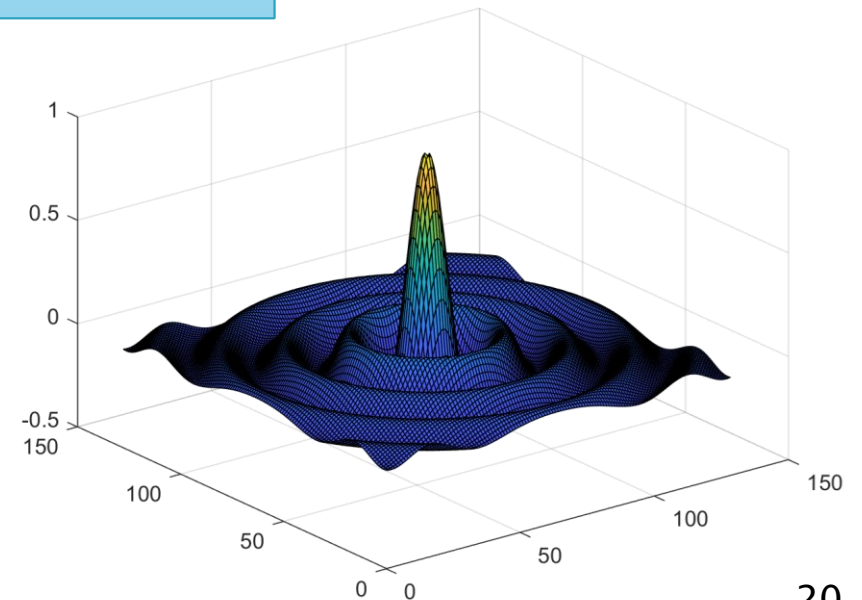
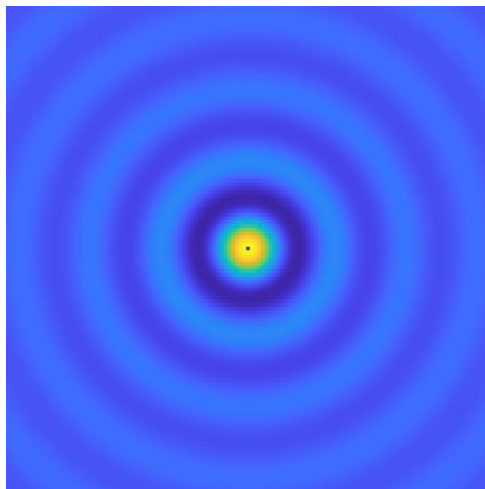
figure, imshow(I)
figure, imshow(I, InitialMagnification=500)
```



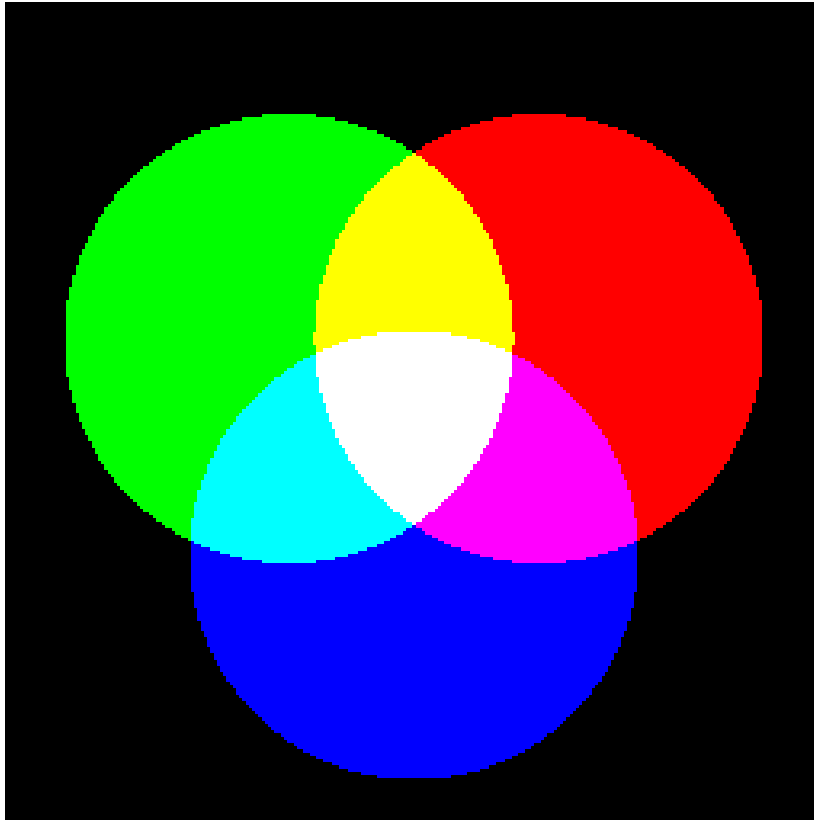
Synthèse analytique en couleurs indexées

```
clear, close all

[X,Y]=meshgrid(-64:64);
R=(X.^2+Y.^2).^0.5;
R=R/3;
I=sin(R)./R;
figure, imshow(I, [], Colormap=parula)
figure, surf(I)
```



Exercice : synthèse additive



- ▶ « Vraies » couleurs
- ▶ Couleurs indexées

« Vraies » couleurs

```
clear, close all

size=255;
radius=70;
dist=45;
[R,G,B]=disks(size,radius,dist);
img=cat(3,R,G,B);
figure, imshow(img)
```

Couleurs indexées

```
clear, close all

size=255;
radius=70;
dist=45;
[R,G,B]=disks(size,radius,dist);
img=uint8(R+2*G+4*B);
map=[0 0 0;1 0 0;0 1 0;1 1 0;0 0 1;1 0 1;0 1 1;1 1 1];
figure, imshow(img, map)
```

Synthèse additive dynamique

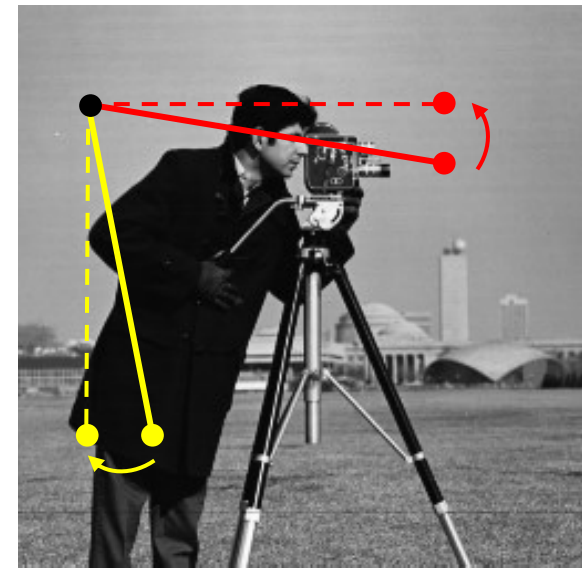
```
clear, close all

size=255; radius=70; dist=45;
[R,G,B]=disks(size,radius,dist);
R=uint8(R*255);
G=uint8(G*255);
B=uint8(B*255);
v=VideoWriter('video.avi','Uncompressed AVI');
v.FrameRate=5;
open(v)
for n=1:99
    q=mod(n,3);
    if q == 0
        img=cat(3,R,G,B);
    elseif q == 1
        img=cat(3,B,R,G);
    else
        img=cat(3,G,B,R);
    end
    writeVideo(v,img);
end
close(v)
```

- ▶ Couleurs circulantes
- ▶ Vidéo
 - 99 images
 - 5 images/seconde
 - sans compression

Exercice : image vs signal

- ▶ 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



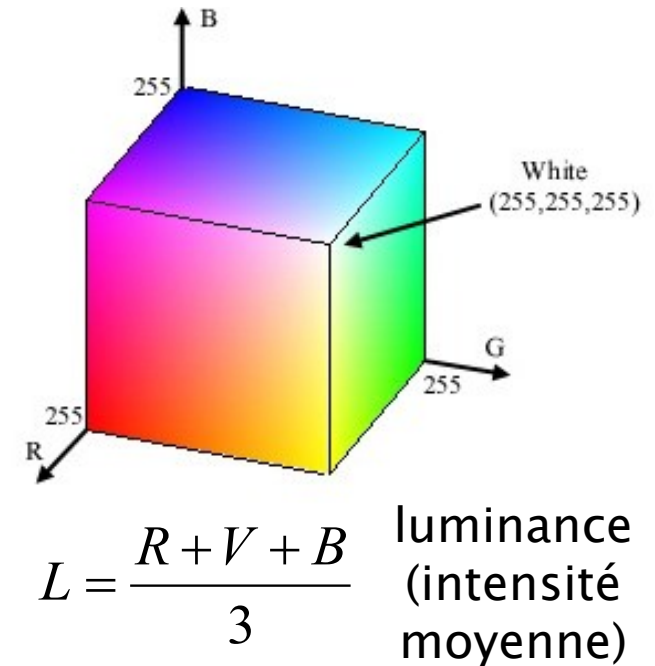
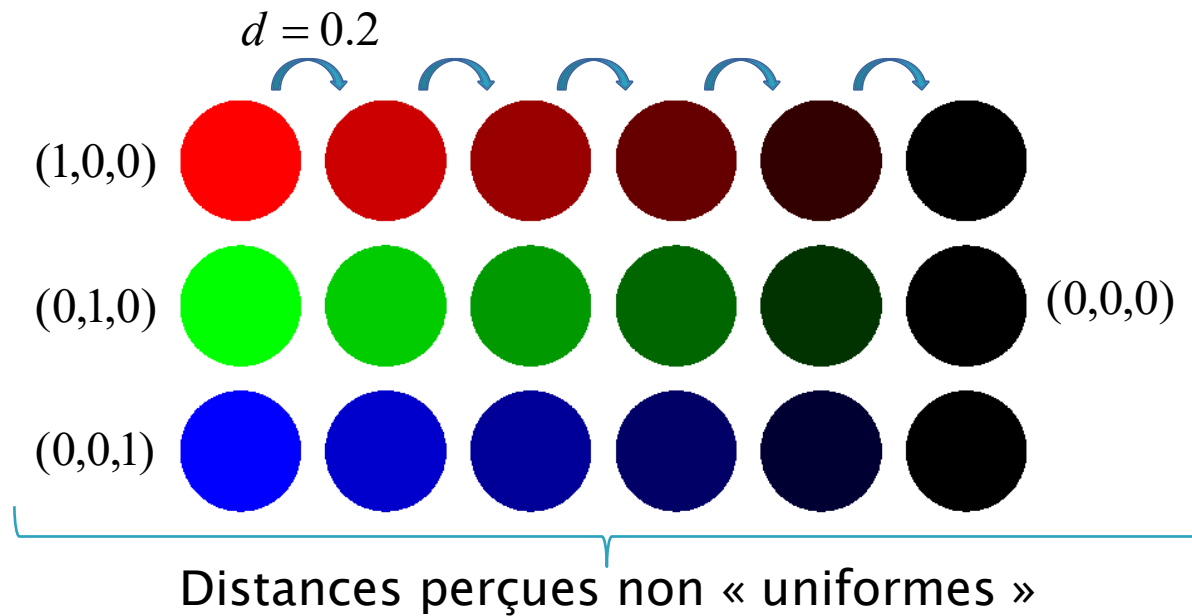
Initiation au Traitement des Images



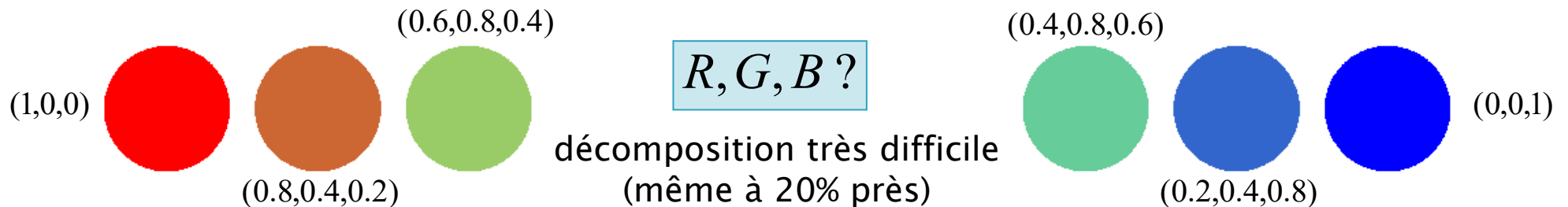
- ▶ Couleur/Amplitude
- ▶ Spatial
- ▶ Fréquentiel

Espaces caractéristiques

Espace chromatique RVB(RGB)



Espace d'acquisition du système visuel humain

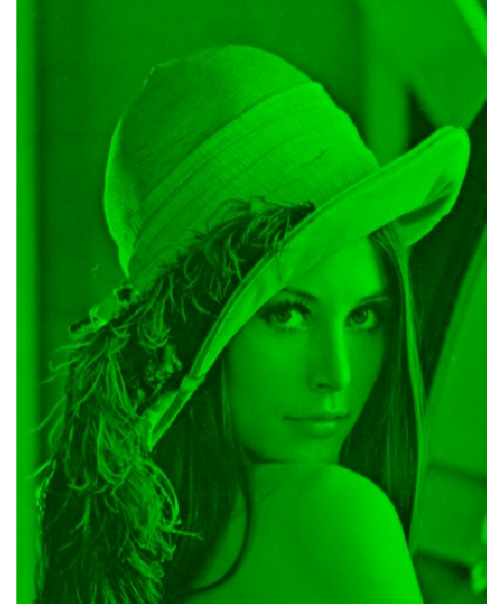
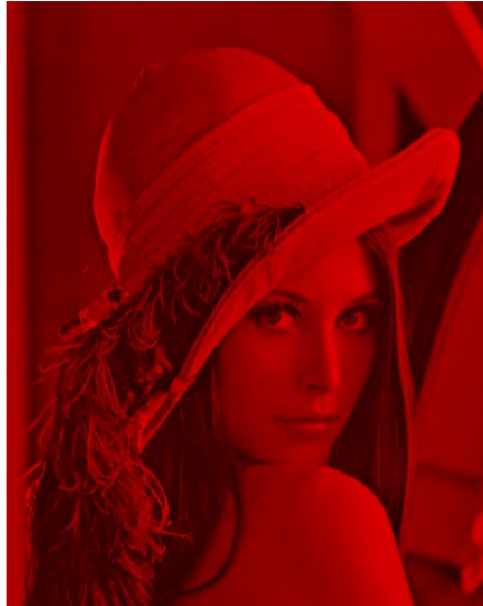


Luminance vs Chrominance



Niveaux de gris

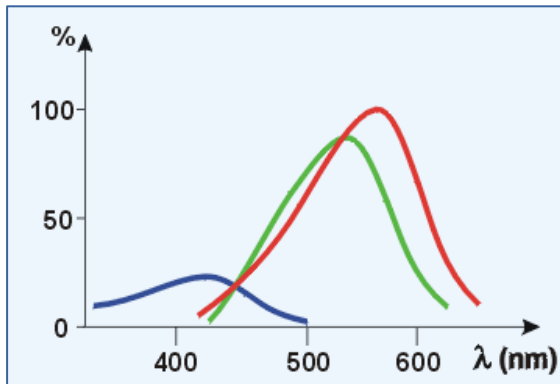
← luminosité plus forte



contrastes plus faibles

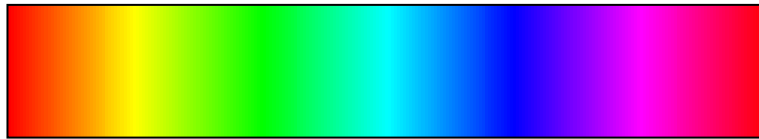


Palettes de primaires pures
(intensités identiques et autres composantes éteintes)



Sensibilité relative des photo-récepteurs de chrominance

Espace chromatique TSL (HSL)



Teinte



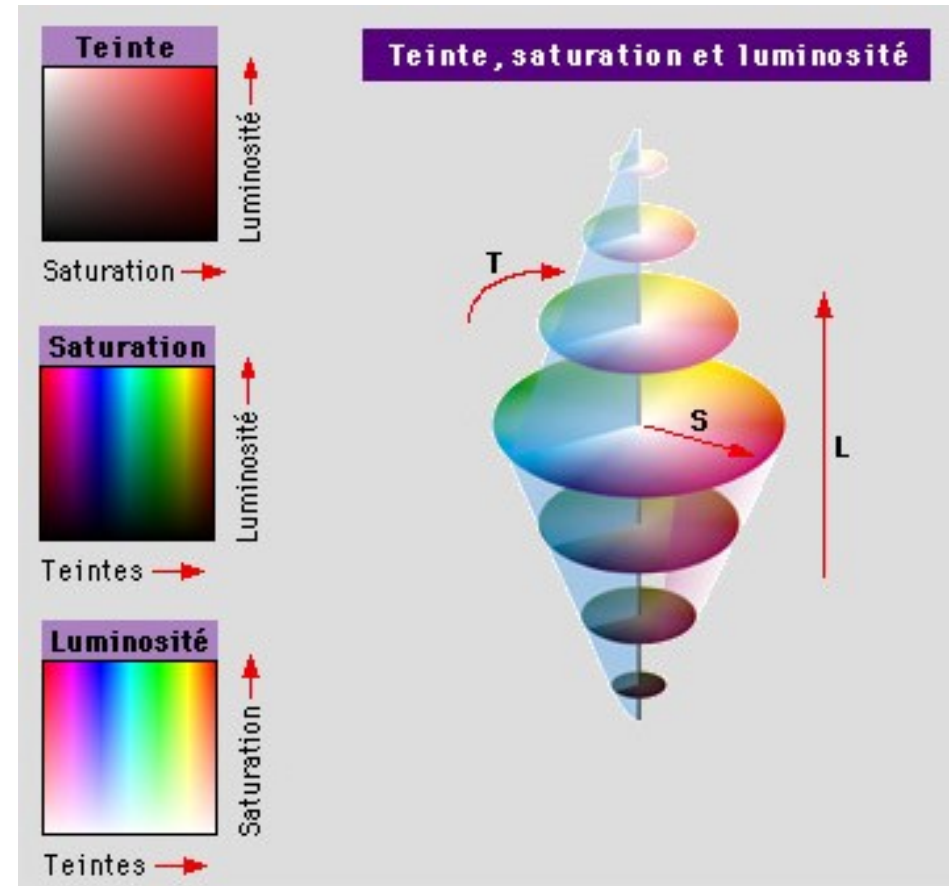
Saturation

« mélange »
avec le gris



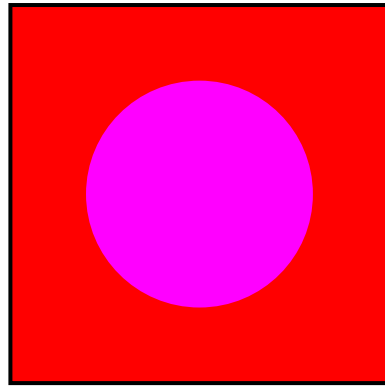
Luminosité

« mélange »
avec le noir

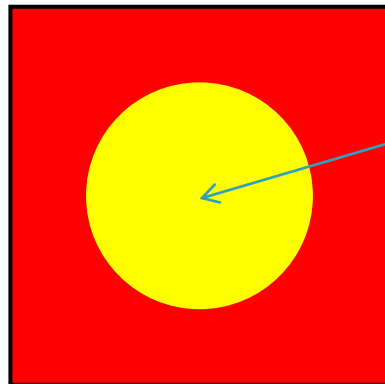


Espace de description naturelle

Luminance « perceptuelle »

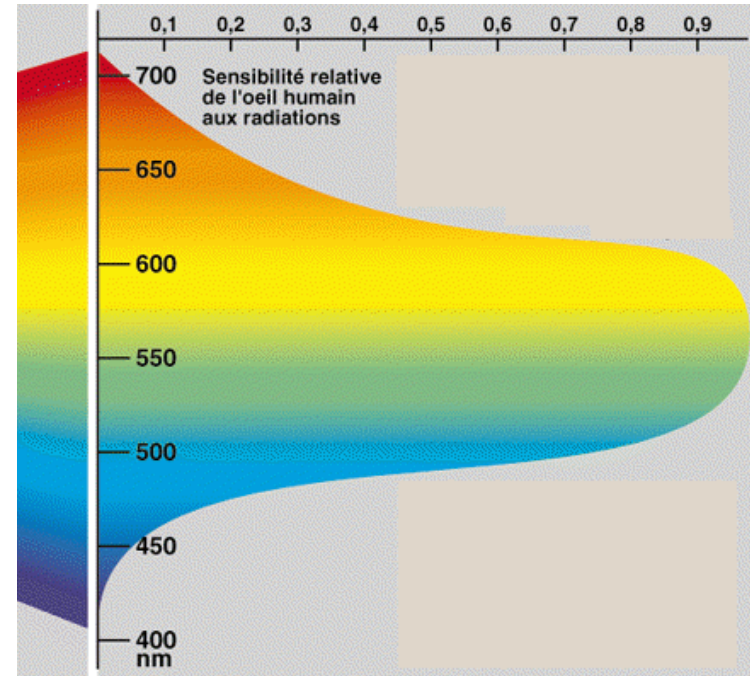


Rouge/Bleu



Rouge/Vert

luminosité
plus forte



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

pondération plus faible

Espace colorimétrique YCbCr

```
clear, 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;
L=(R+G+B)/3;

figure, imshow(uint8(A))
figure, imshow(uint8(L))
figure, imshow(uint8(Y))
```

```
tx=90:350;
ty=160:330;

figure,
subplot(2,3,1)
imshow(uint8(R(ty,tx)))
subplot(2,3,2)
imshow(uint8(B(ty,tx)))
subplot(2,3,3)
imshow(uint8(G(ty,tx)))
subplot(2,3,4)
imshow(uint8(Cr(ty,tx)))
subplot(2,3,5)
imshow(uint8(Cb(ty,tx)))
subplot(2,3,6)
imshow(uint8(A(ty,tx,:)))
```



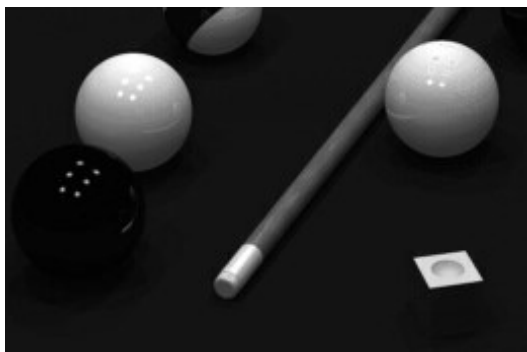
L



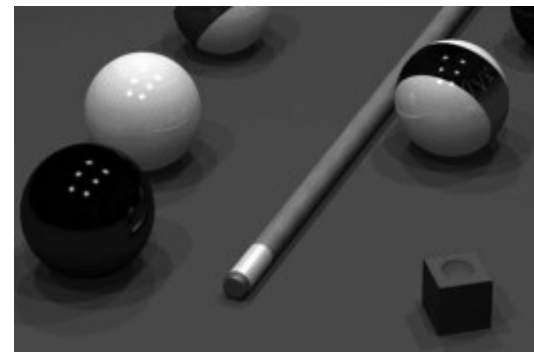
Y



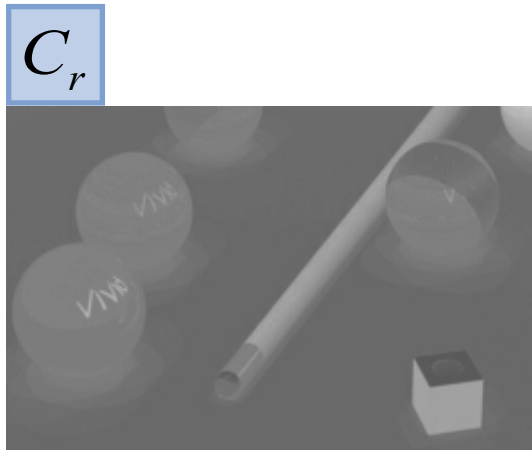
R



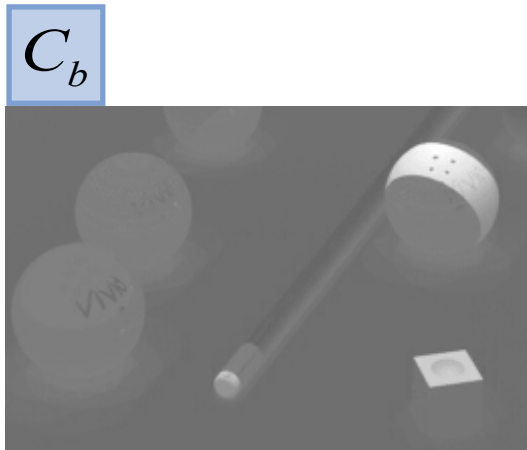
B



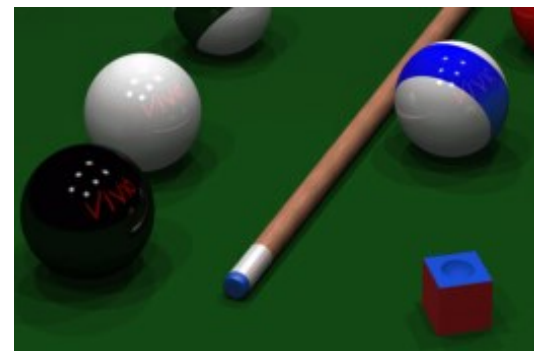
G



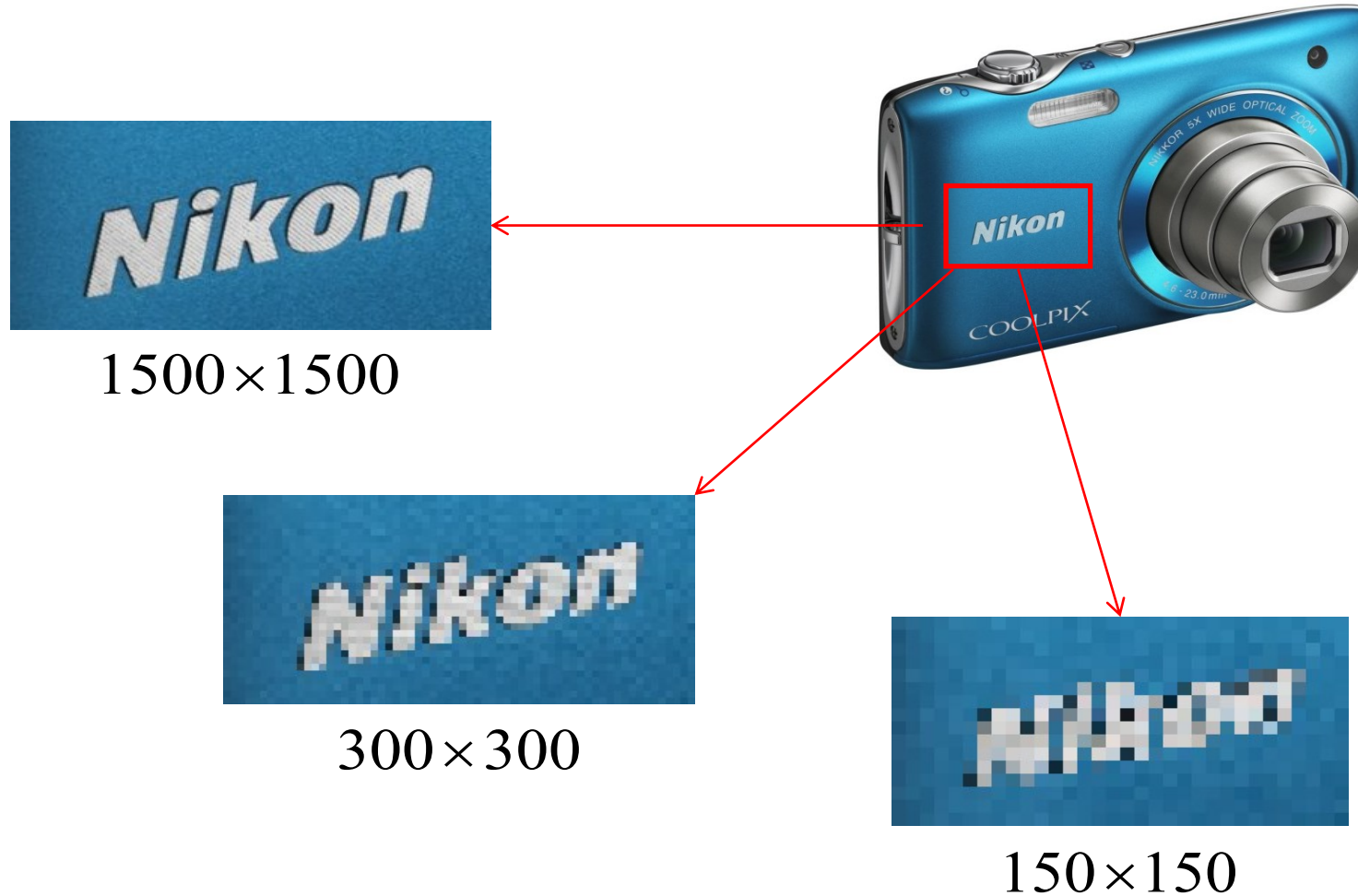
C_r



C_b



Résolution spatiale



```
clear, close all

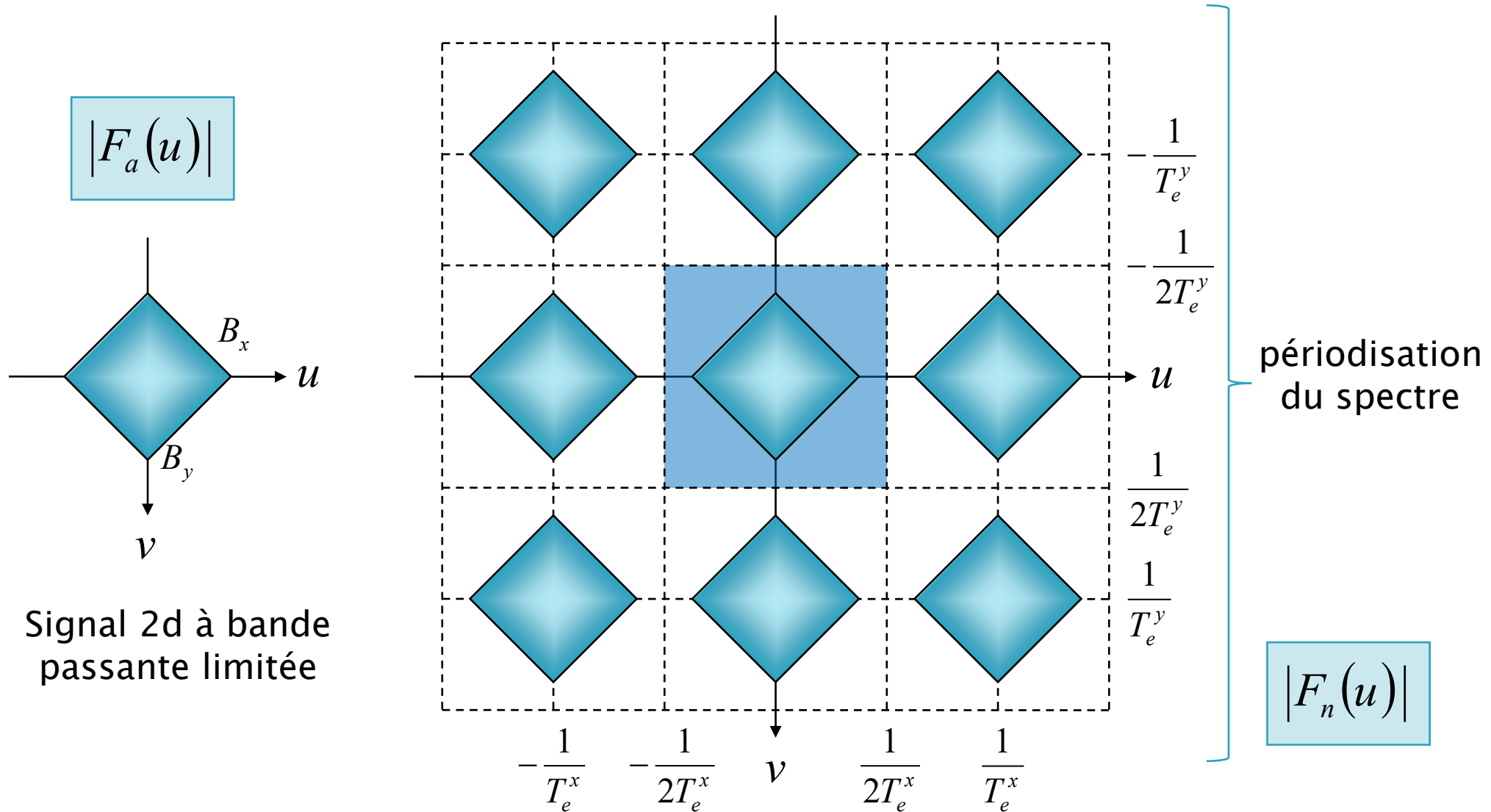
A=imread('pcb.jpg');

figure, imshow(A)
Ae=A(81:240,481:600);
figure, imshow(Ae, InitialMagnification=400)

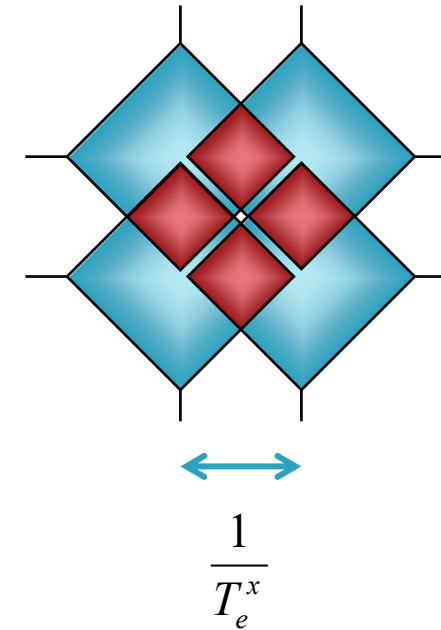
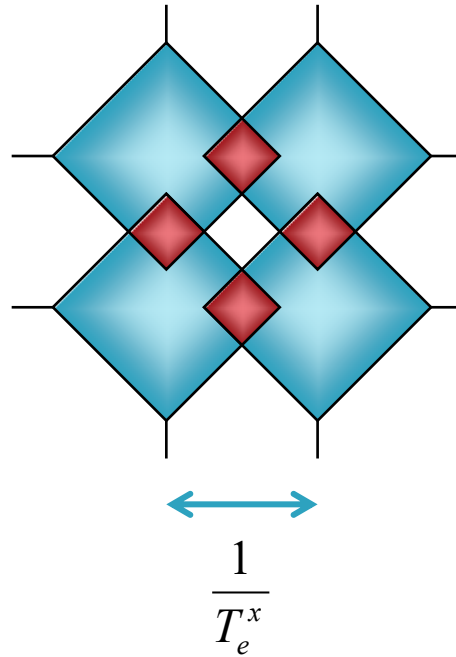
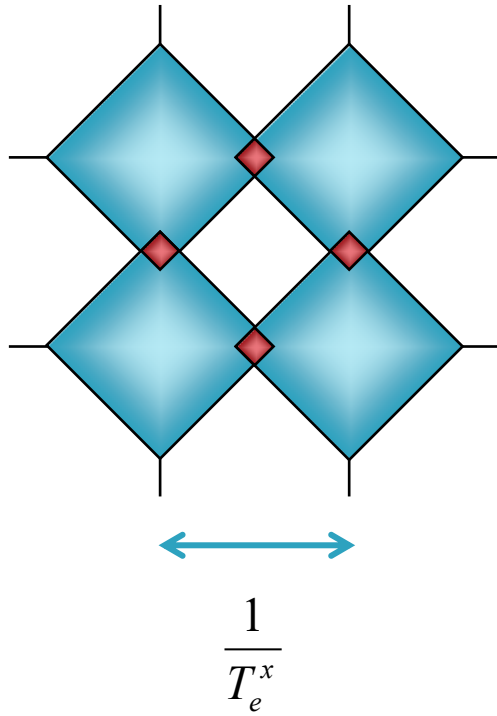
[Ny,Nx]=size(A);
B=A(1:4:Ny,1:4:Nx);

figure, imshow(B, InitialMagnification=400)
Be=B(21:60,121:150);
figure, imshow(Be, InitialMagnification=1600)
```


Spectre d'une séquence discrète 2d



Recouvrement fréquentiel



Des hautes fréquences ... aux basses fréquences

Transformée de Fourier

Transformée directe

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

variables
fréquentielles

variables
spatiales

**Extension au cas des
signaux bidimensionnels
(images)**

Transformée inverse

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

Transformée de Fourier discrète 2d

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

Transformée (continue)
de Fourier (TF)

échantillonnage spatial

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

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

transformée
(f_e^x, f_e^y)-périodique

périodes
d'échantillonnage

fenêtrage et
échantillonnage des fréquences

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

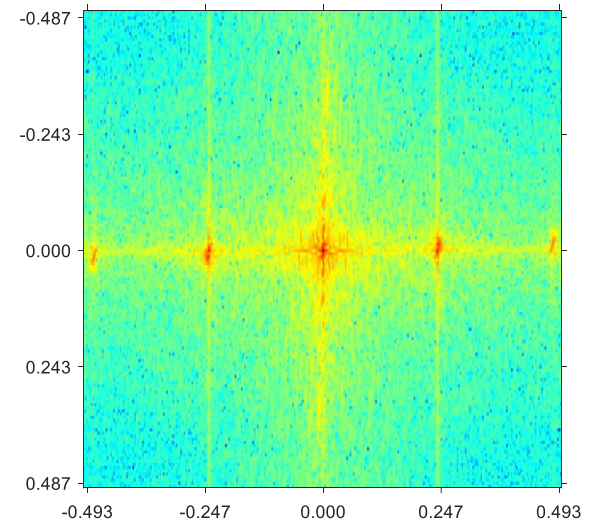
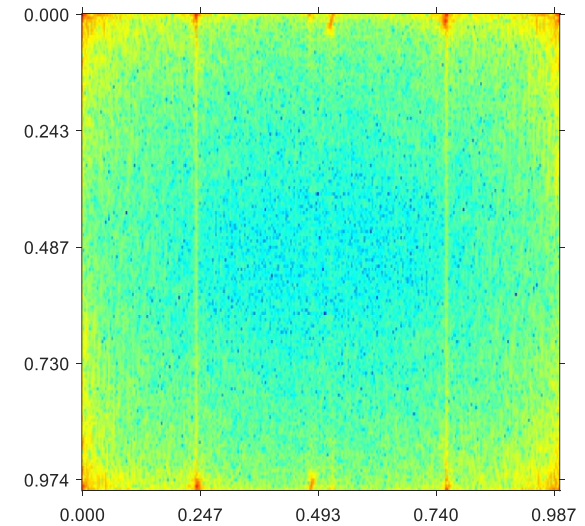
$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi\left(k\frac{m}{M} + l\frac{n}{N}\right)}$$

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

TFD

```
clear, close all

I=imread('building.jpg');
figure, imshow(I)
If=log10(abs(fft2(I)));
figure, draw_dft(If, 0)
figure, draw_dft(If, 1)
```




```

function []=draw_dft(If,shift)
if shift
    If=fftshift(If);
end
imshow(If/max(max(If)), Colormap=jet), axis square, axis on
[ux, lx]=ticksDim(size(If,2),2,shift);
[uy, ly]=ticksDim(size(If,1),2,shift);
xticks(ux), xticklabels(lx)
yticks(uy), yticklabels(ly)
end

function [u,l]=ticksDim(s,N,shift)
ds=floor((s-1)/(N*2));
f=(0:s-1)/s;
u=1+(0:N*2)*ds;
if shift
    f=f-floor(s/2)/s;
    u=u+floor(s/2)-N*ds;
end
l=arrayfun(@(x) num2str(x, '%.3f'), f(u), 'UniformOutput', false);
end

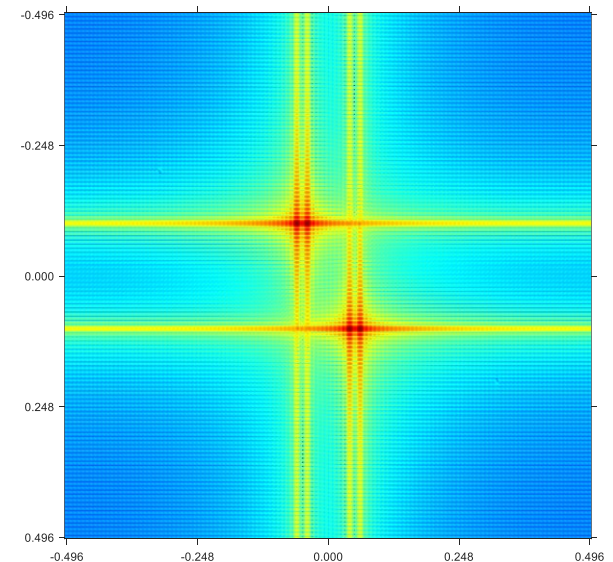
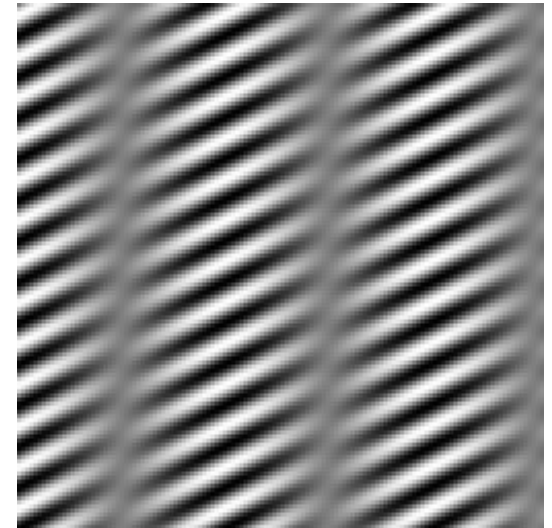
```

draw_dft.m

TFD et fenêtrage

```
clear, 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, imshow(I, [])
If=log10(abs(fft2(I,512,512)));
figure, draw_dft(If,1)
```

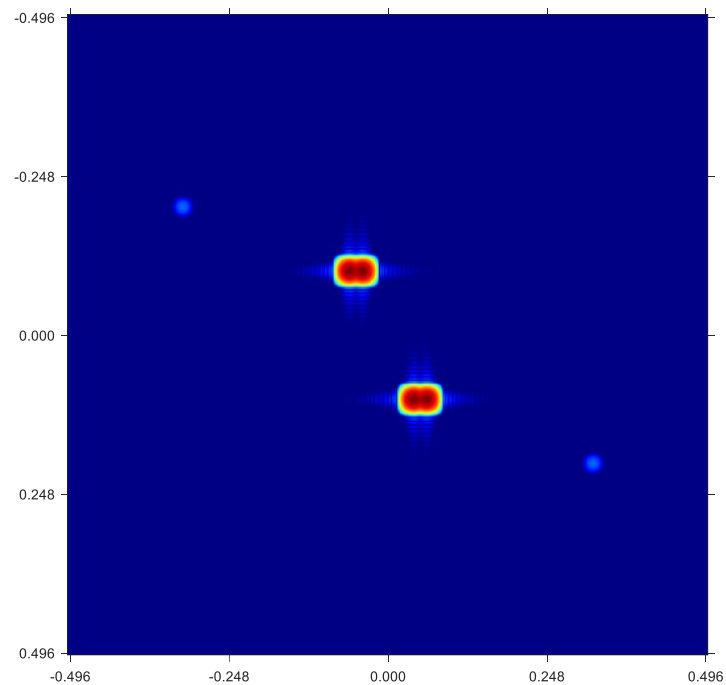
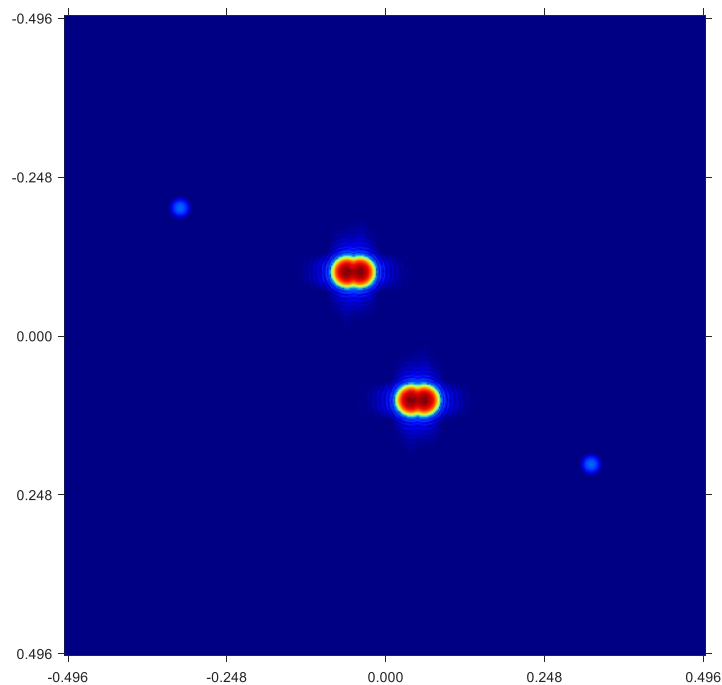
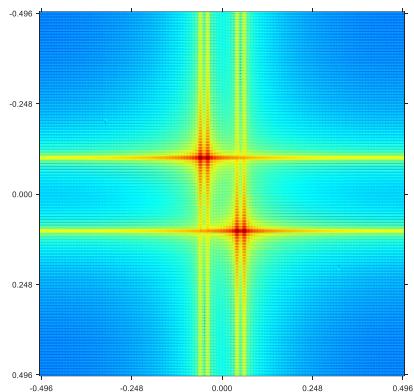


Kaiser séparable

```
W=kaiser(128,10)*kaiser(128,10)';  
Iw=I.*W;  
If2=log10(abs(fft2(Iw,512,512)));  
figure, draw_dft(If2,1)
```

```
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=log10(abs(fft2(Iw2,512,512)));  
figure, draw_dft(If3,1)
```

Kaiser à symétrie circulaire



**Compromis
détection/localisation**

Initiation au Traitement des Images

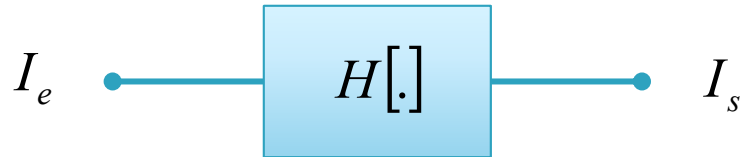


- ▶ Filtrage linéaire
 - Filtres RIF
 - Détection de contours
- ▶ Filtrage non linéaire

Traitements

Filtrage linéaire

Système linéaire



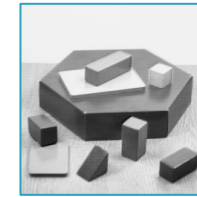
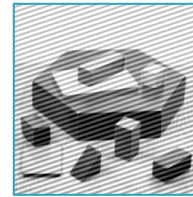
entrée et sortie reliées par un produit de convolution

$$I_s = I_e * H$$

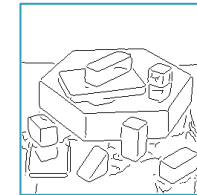
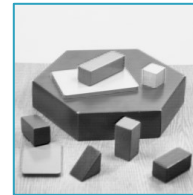
caractérisation par la réponse impulsionnelle

$$h(t) = T[\delta(t)]$$

impulsion



débruitage



contours

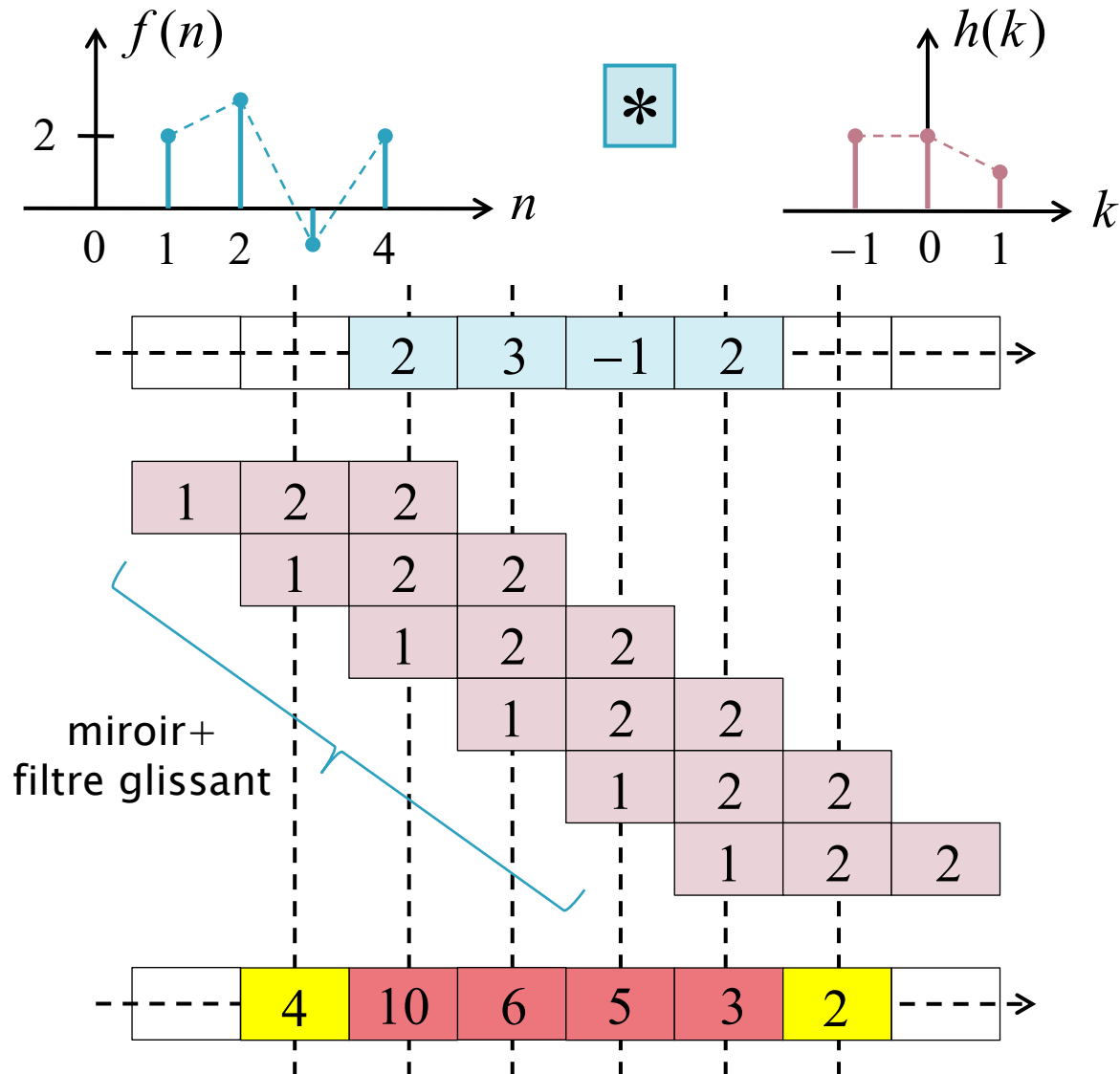
Filtre linéaire RIF

Application d'un filtre linéaire discret à réponse impulsionnelle finie

défini sur $(2K+1) \times (2L+1)$ échantillons

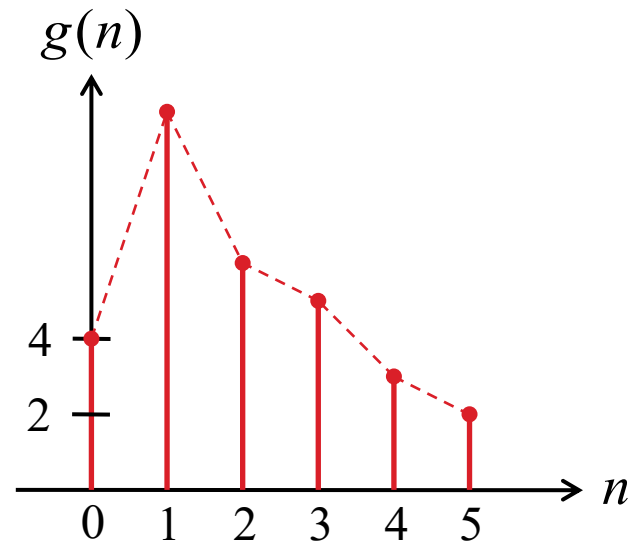
$$g(m, n) = \sum_{k=-K}^K \sum_{l=-L}^L f(m-k, n-l) h(k, l)$$

Convolution discrète 1d

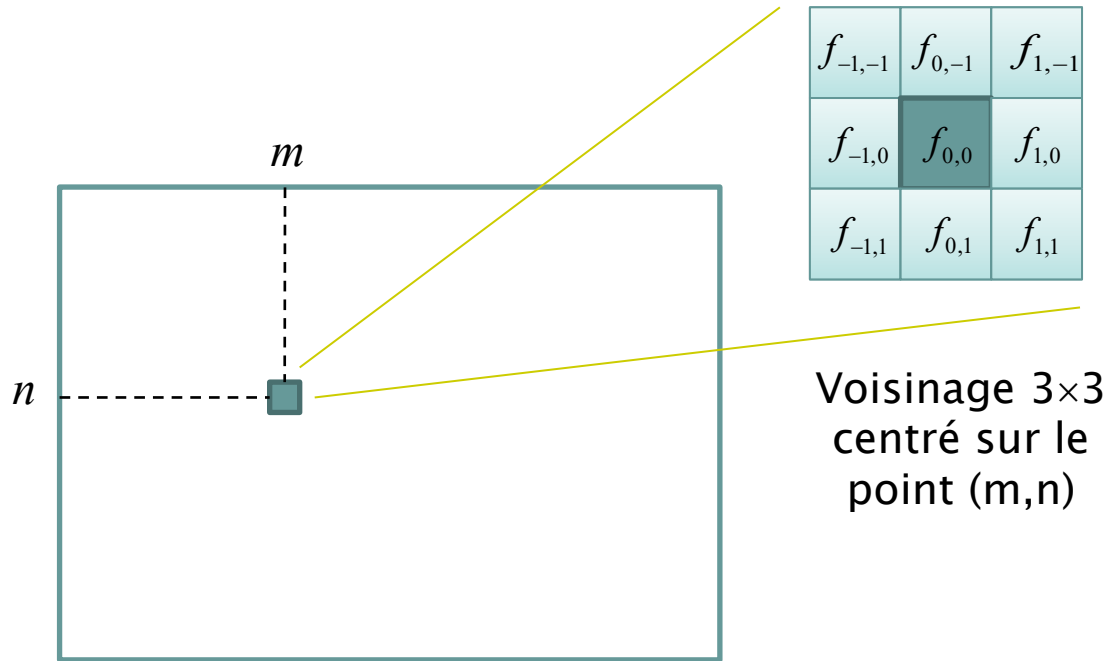


$$g(n) = \sum_{k=-K}^K f(n-k)h(k)$$

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



Convolution discrète 2d



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

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

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

Example : convolution 3x3

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

I

*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

H

```
clear
close all

I=[3 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')
Ic3=conv2(I,H,'valid')
```

I =

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

Ic1 =

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

Ic2 =

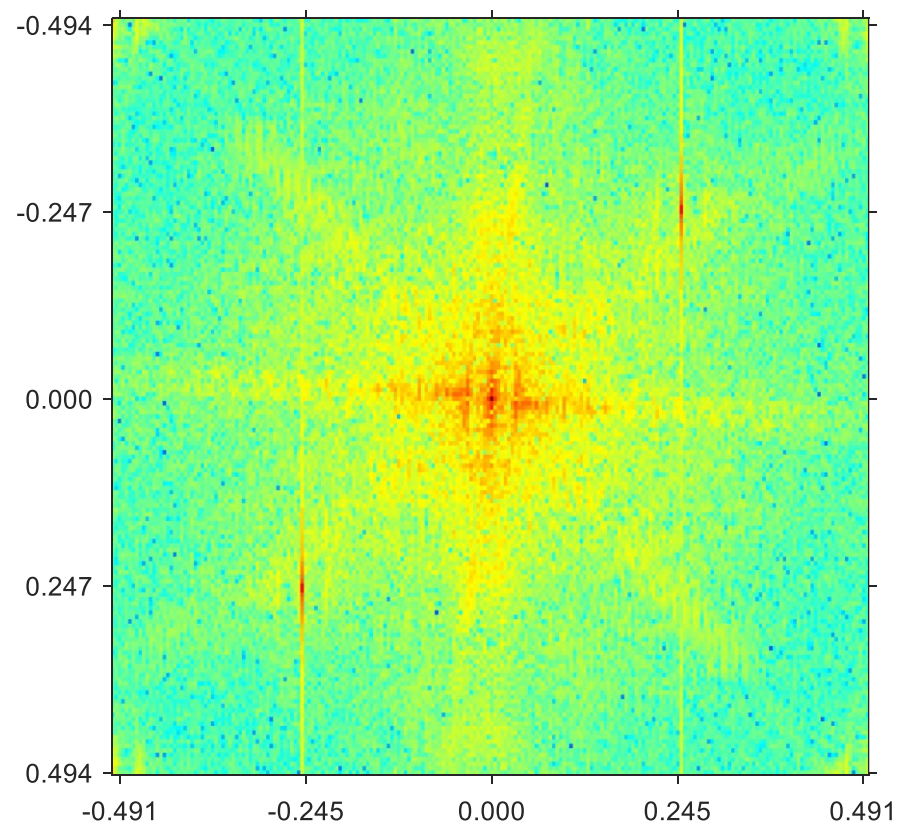
0.6667	0.8889	0.6667	0.6667	0.4444
0.8889	1.2222	1.0000	1.0000	0.6667
0.6667	1.0000	1.0000	1.0000	0.6667
0.4444	0.6667	0.6667	0.6667	0.4444

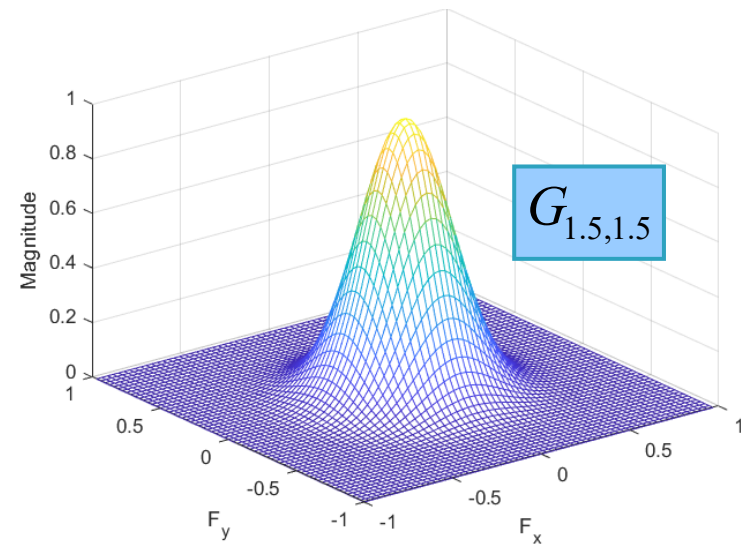
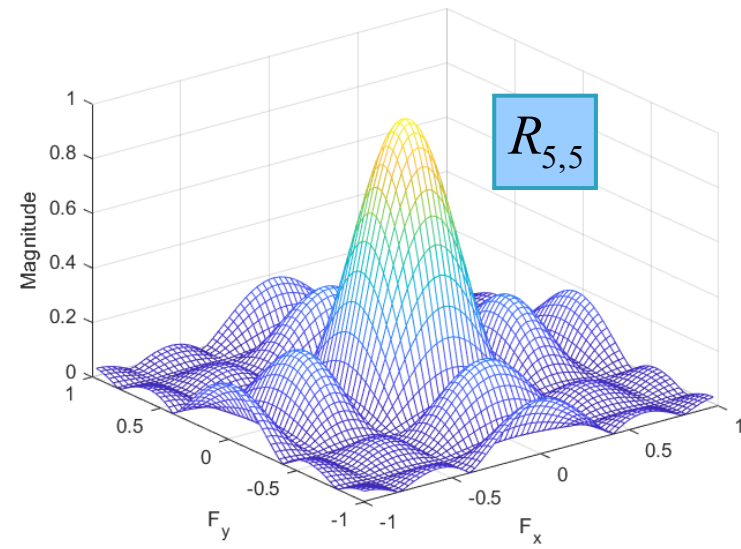
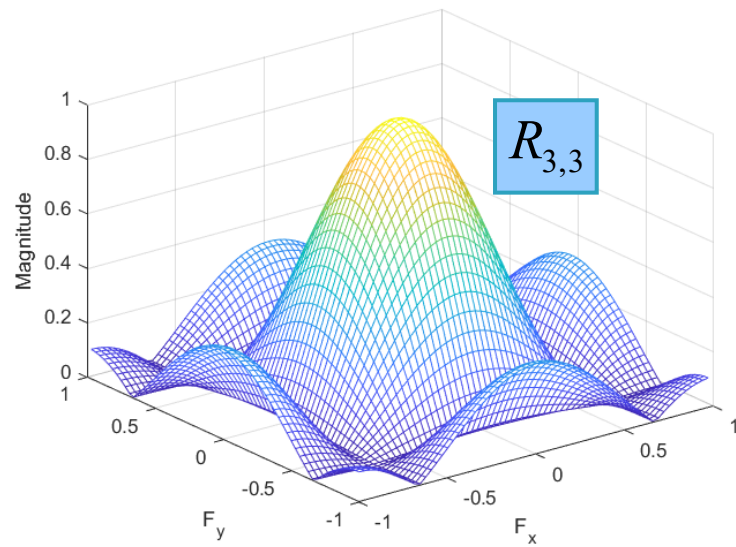
Ic3 =

1.2222	1.0000	1.0000
1.0000	1.0000	1.0000

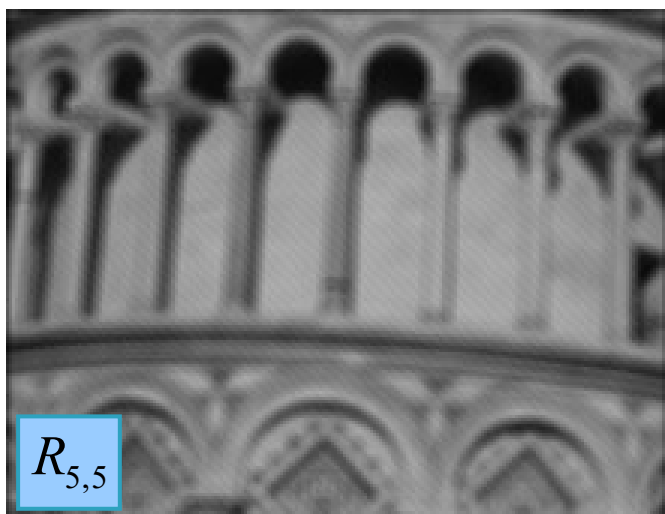
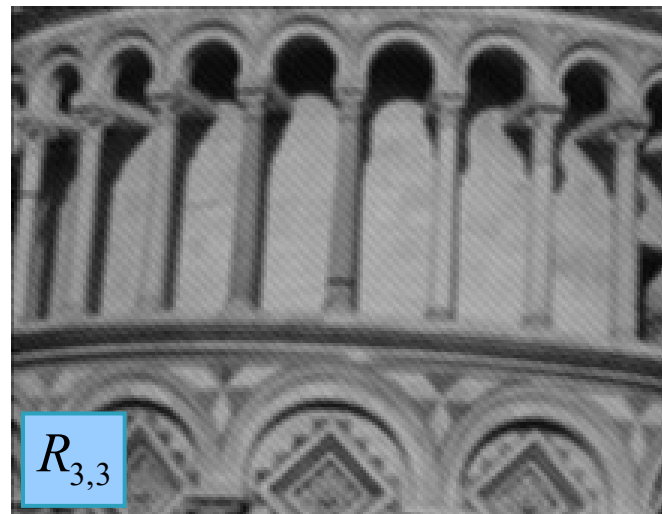
Exercice : détramage

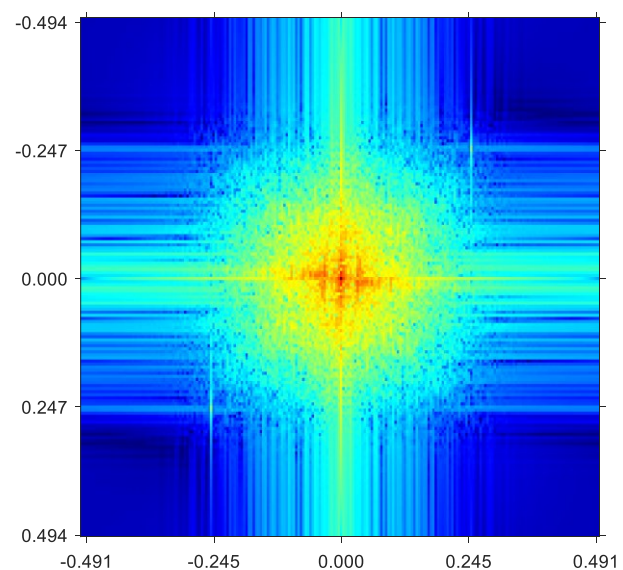
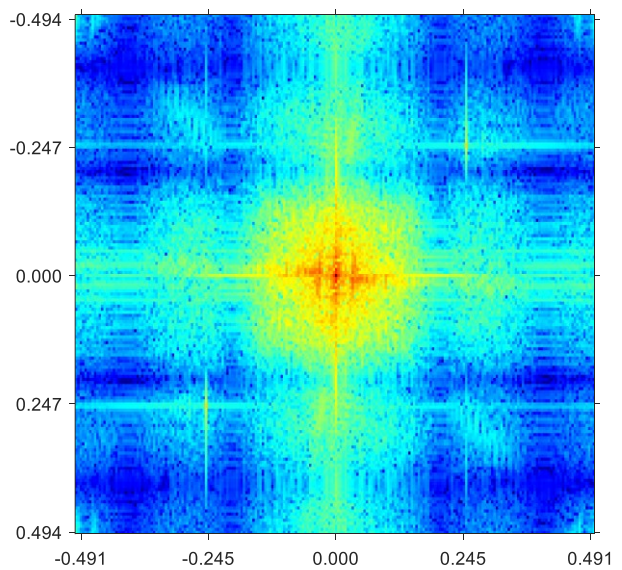
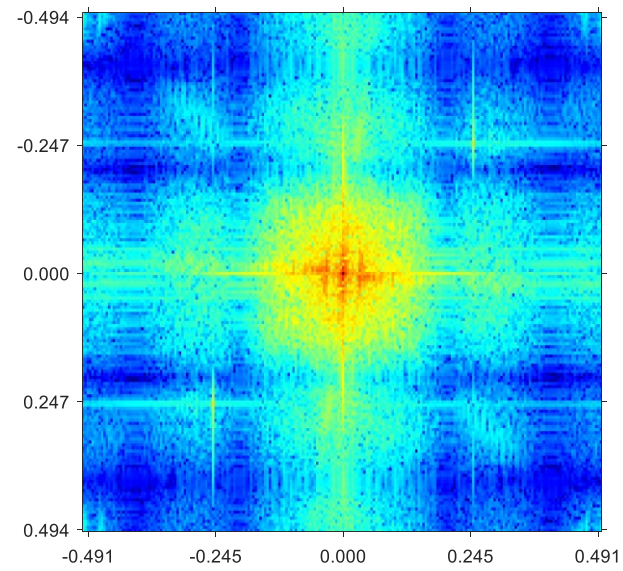
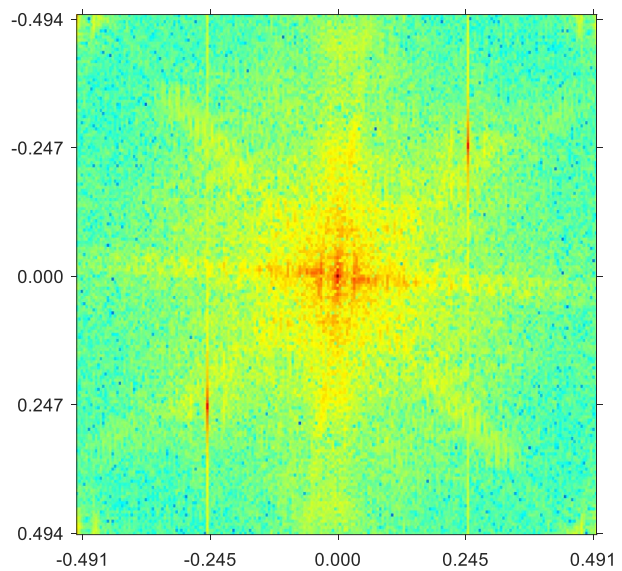
- ▶ Affichage de l'image 'pise_ext.bmp'
- ▶ Filtrage passe-bas pour éliminer/réduire le bruit
 - Justification du choix (visualisation de la TFD)
 - Vérification de l'adéquation des filtres (freqz2)
- ▶ Affichage des images filtrées et de leur spectre
- ▶ Calcul et affichage du « bruit » (différence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

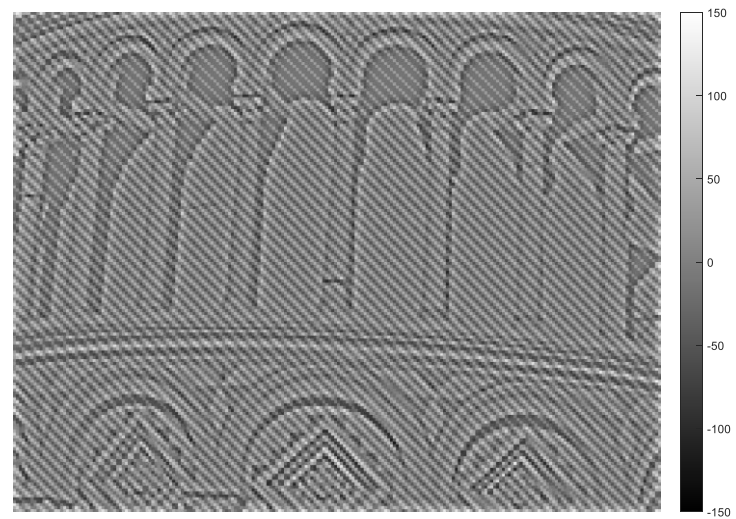
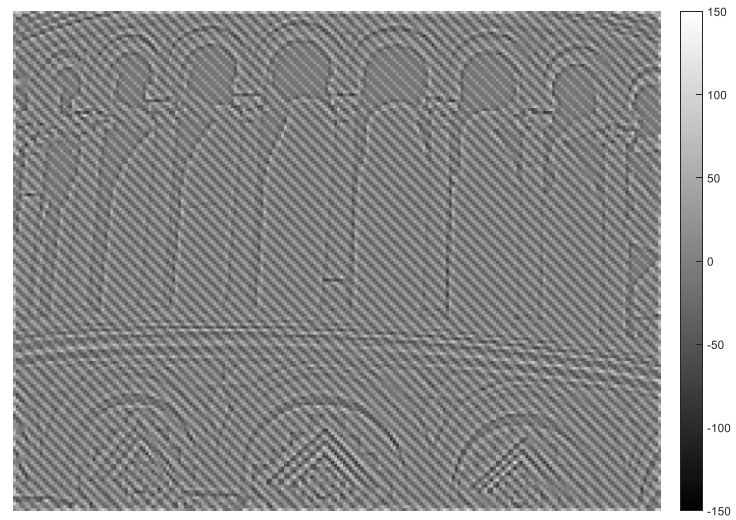
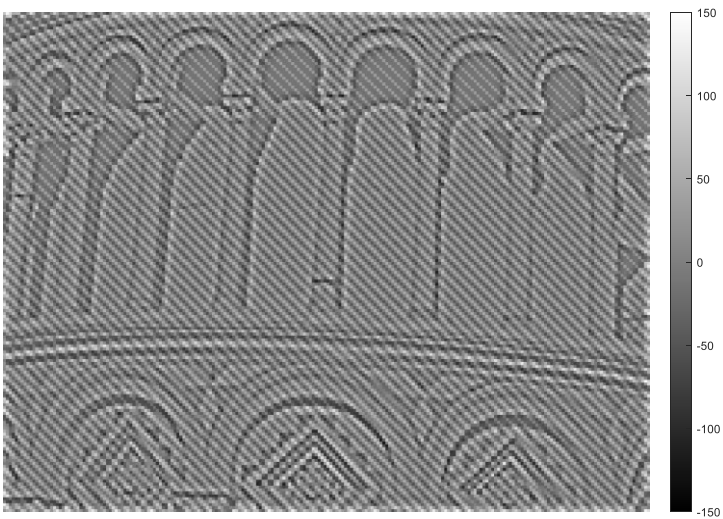




```
B=conv2 (A,H1,'same');  
figure, imshow(uint8(B));  
C=conv2 (A,H2,'same');  
figure, imshow(uint8(C));  
D=conv2 (A,H3,'same');  
figure, imshow(uint8(D));  
  
figure, compute_draw_dft(B)  
figure, compute_draw_dft(C)  
figure, compute_draw_dft(D)  
  
figure, imshow(A-B, [-150 150]), colorbar  
figure, imshow(A-C, [-150 150]), colorbar  
figure, imshow(A-D, [-150 150]), colorbar
```







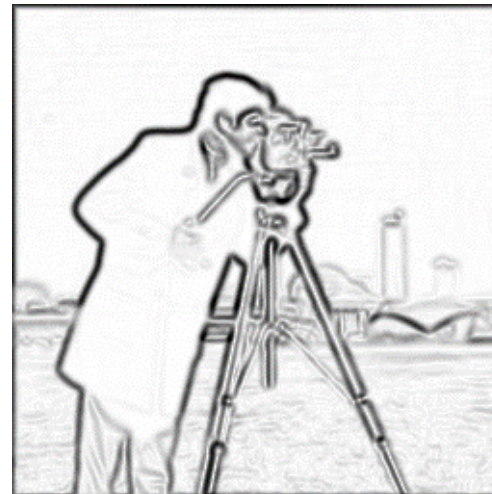
Détection de contours

```
clear, close all

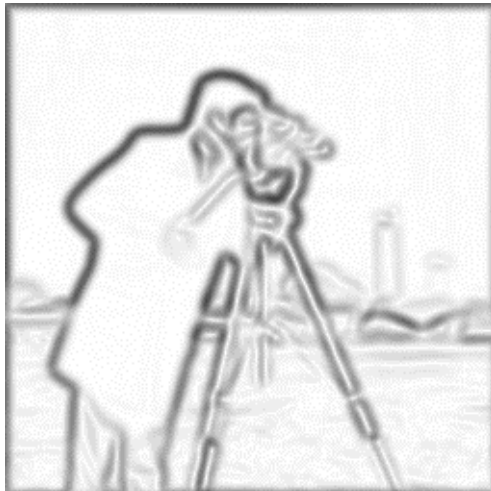
A=double(imread('cameraman.tif'));
figure, imshow(uint8(A))
sigma=0.5;
[X,Y]=meshgrid(-2:2);
H=exp(-(X.^2+Y.^2)/(2*sigma^2));
Hx=-X.*H;
Hx=Hx/sum(sum((-X.*Hx)));
Hy=Hx';
Gx=conv2(A,Hx,'same');
Gy=conv2(A,Hy,'same');
G=(Gx.*Gx+Gy.*Gy).^0.5;
figure, imshow(G, [0 50])
colormap(flipud(gray(256)))
```



$\sigma = 0,5$



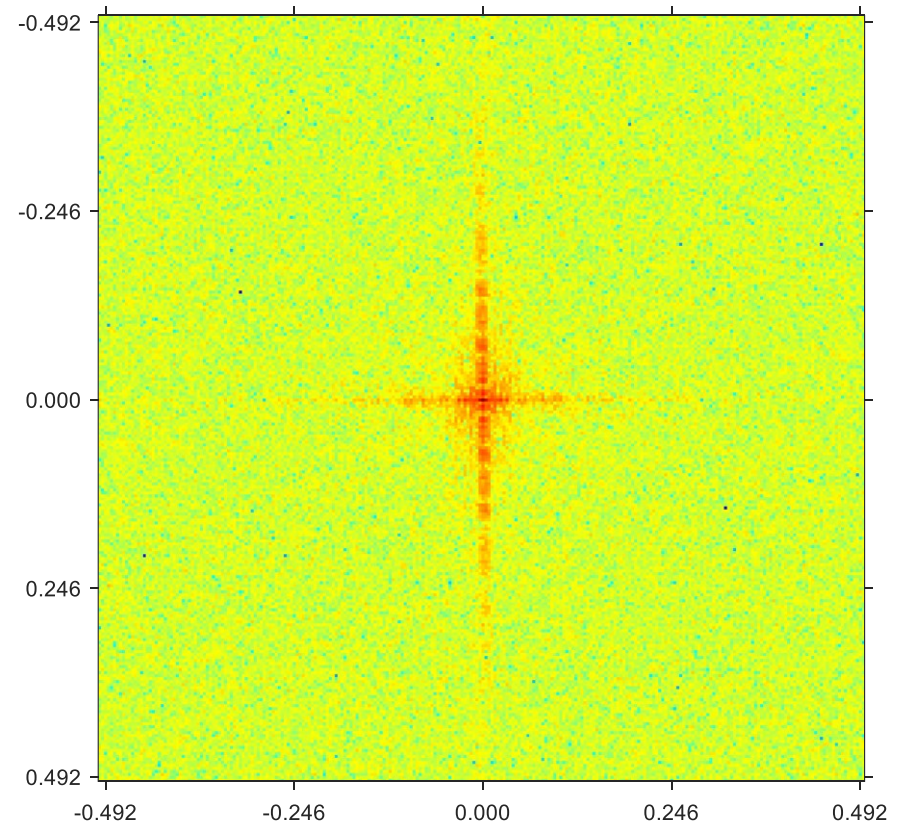
$\sigma = 1,5$

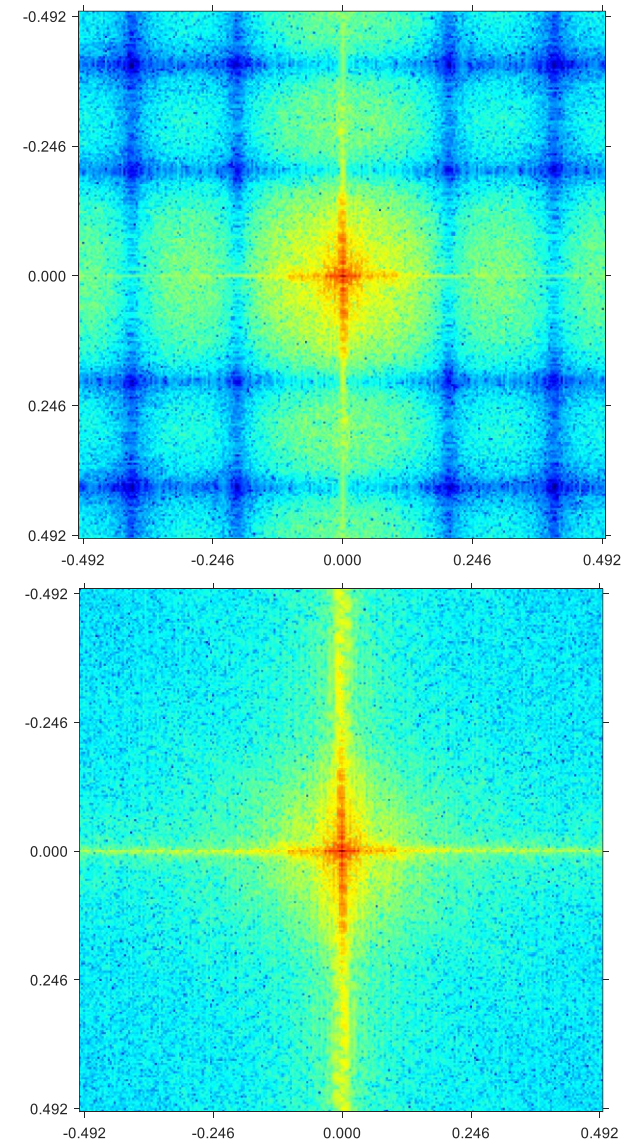


$\sigma = 2,5$

Exercice : filtrage d'ordre

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





Syntaxes utiles

Initialisation

```
x=2;  
u=[5 -3 2 4];  
A=[1 2 3 4;5 6 7 8;...  
   9 10 11 12];
```

Accès

```
y=u(2);  
B=A(2:end,1:2);  
B=A(3,:);
```

Réorganisation

```
t=A(:);
```

Opérations

```
v=u+x;  
v=u+u;  
v=u.*u;  
v=u>1;
```

```
y=u*u';
```

```
A=u'*u;  
B=A+x;  
B=A*u';  
B=A+A;  
B=A*A';  
B=A.*A;  
B=A>2;
```

Fonctions utiles

whos, size

zeros, ones

double, uint8

imread, imwrite

flipud, cat

figure, subplot, hold, axis

imshow, colorbar

plot, line, surf

Xticks, yticks

Xticklabels, yticklabels

load, save

ginput

min, max, abs, log10, sqrt, mod

fix, floor

linspace, meshgrid

fft2, fftshift

conv2, freqz2

medfilt2

VideoWriter, writeVideo, open, close