Filière Electronique - 2ème année

Initiation au Traitement des Images

Organisation

- Cours (8h)
  - Evaluation : examen écrit sans documents
  - Intervenant : Y. Berthoumieu
- Enseignement intégré (8h)
  - Cours et mise en oeuvre dans l'environnement Matlab
  - Intervenants : G. Bourmaud/J. Daniel/M. Donias (en 1/4 de promo)
- (TP, 8h, UV Optionnelle TSI)
  - Chroma-Keying, Détramage, Contours/Rehaussement
  - Evaluation : rapport (par binôme)
  - Intervenant : M. Donias

Sources d'images numériques

Acquisition analogique

CAN

Acquisition numérique

Simulation/infographie

Domaine du non visible

Définition (1 / 3)

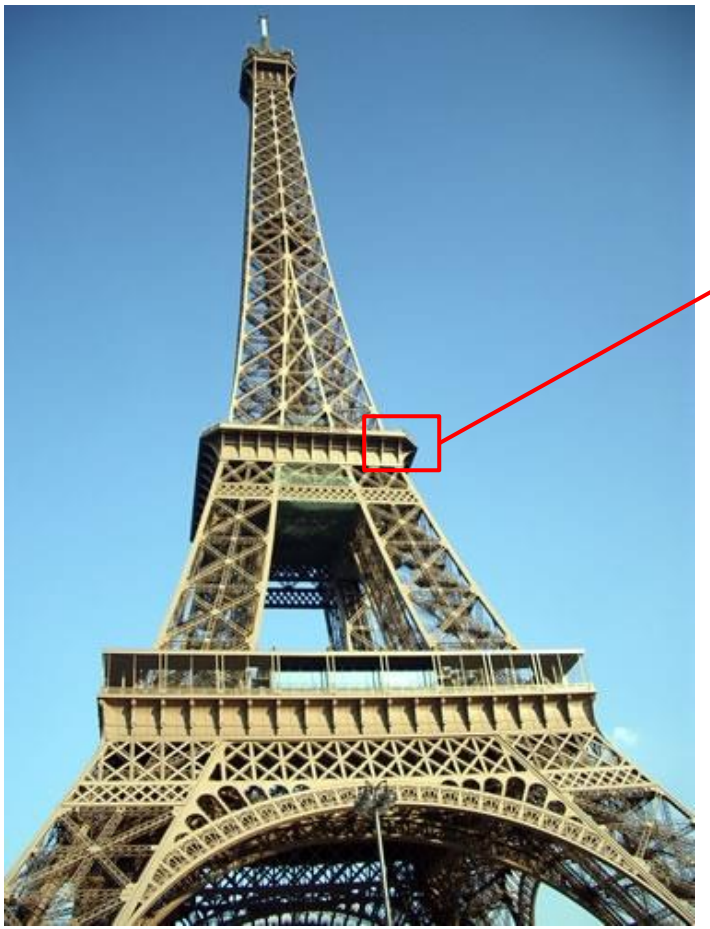


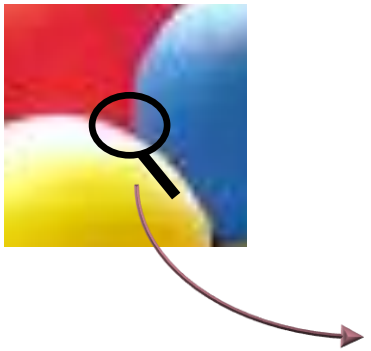
Image numérique

=

Matrice de pixels « colorés »

- Dimensions (nombre de pixels)
- Coordonnées (position du pixel)
- Valeur (couleur du pixel)

Définition (2/3)



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

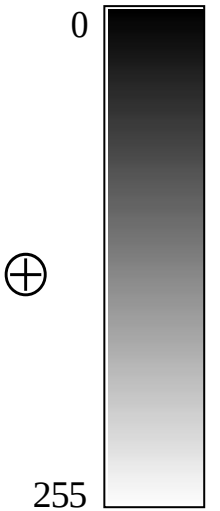
⊕

Format usuel : entiers 0 à 255

Définition (3/3)

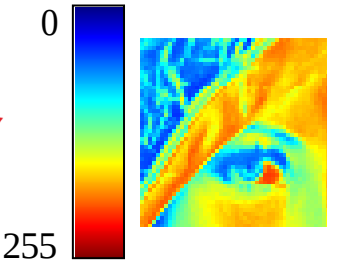
60	64	62	62	55	51
59	73	98	90	66	54
72	105	167	170	120	74
88	91	188	202	184	150
103	77	191	205	203	190
75	131	208	209	207	202

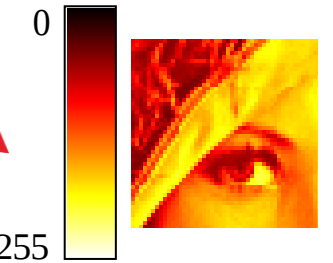
Intensité scalaire

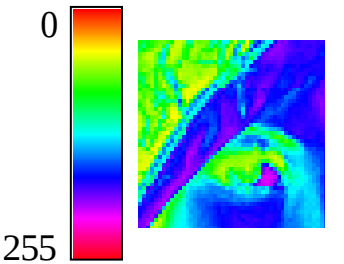


Palette
(niveaux de gris)

Autres palettes







Couleurs indexées ou « fausses » couleurs

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

Modélisation

spectre émis

$I(x, y, z, t)$

Lumière naturelle

Sources lumineuses

Lampe halogène

Tube fluorescent

réflectivité (diffuse, spéculaire, …)

Objet (illuminé)

$R(x, y, z, t)$

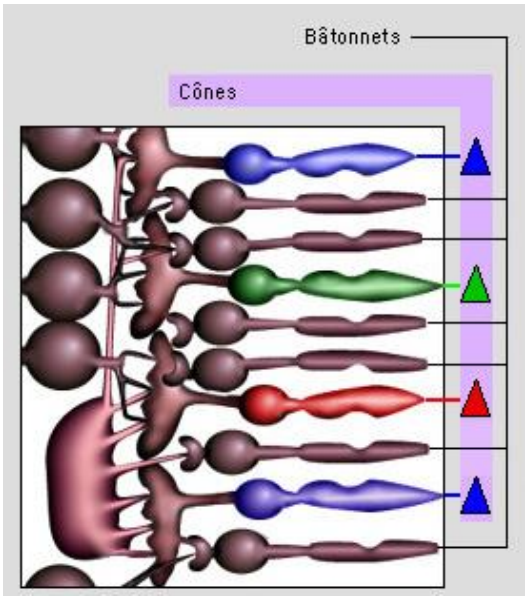
spectre réfléchi

Observateur

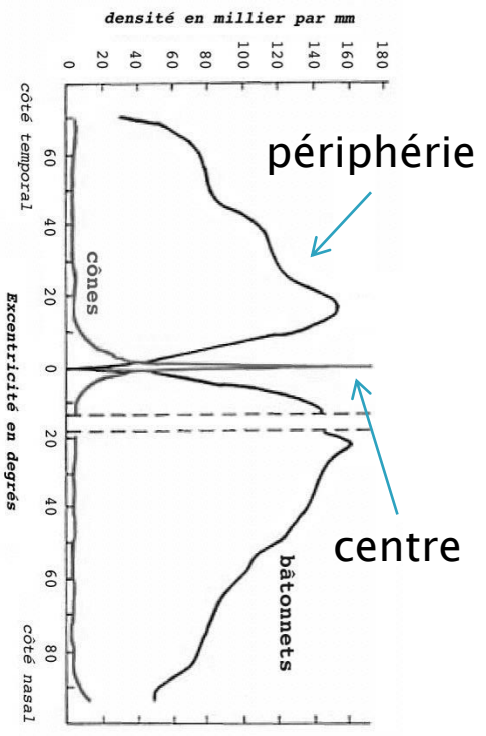
couleur « perçue »

- Source
- Objet
- Milieu de propagation

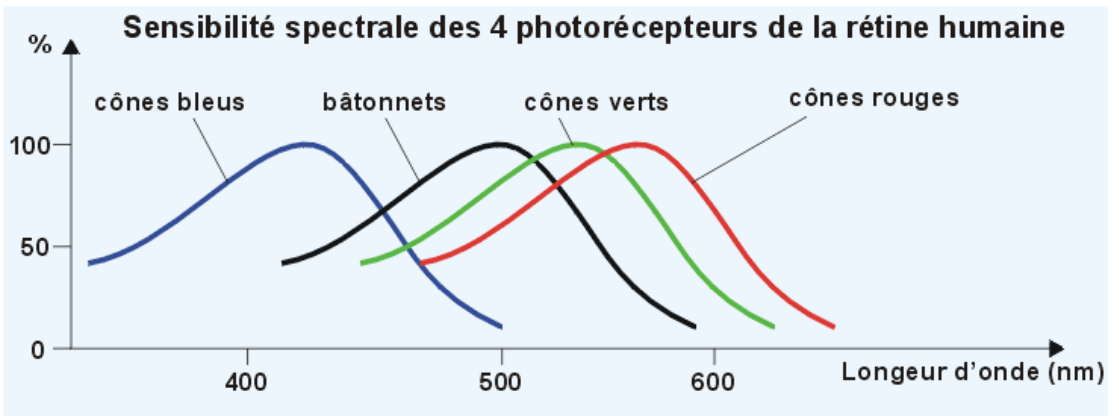
Système visuel humain (SVH)



- Bâtonnets
  - ✓ ~ 120 millions, forme allongée
  - ✓ En périphérie de la rétine
  - ✓ Vision nocturne (regroupés, très sensibles à la luminosité mais pas aux détails)
- Cônes
  - ✓ ~ 6 millions, 3 sous-familles (S, M, L)
  - ✓ Au centre de la rétine (fovéa)
  - ✓ Vision diurne (densité élevée)



Distribution angulaire

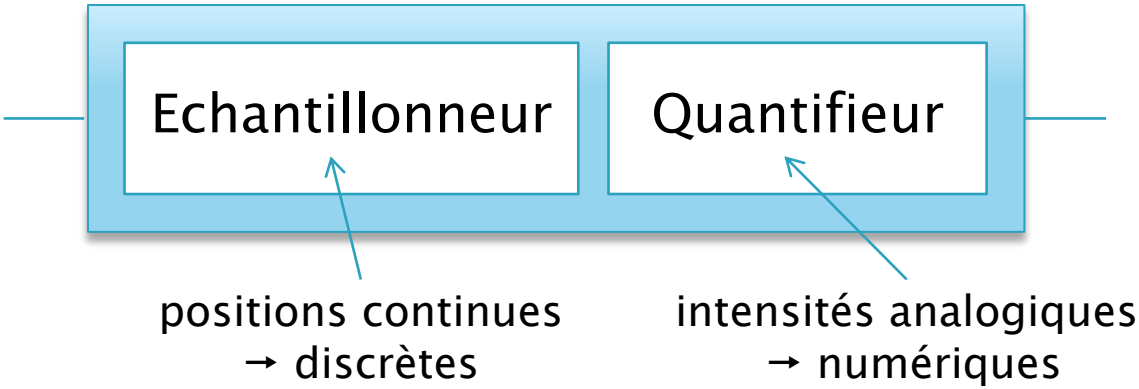


Système Visuel Humain (SVH)

=

Système trichromate

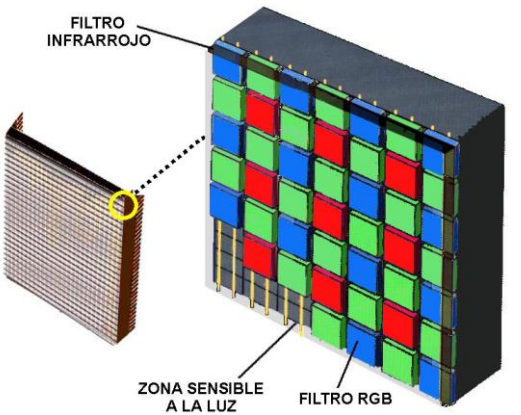
Systèmes d'acquisition des couleurs



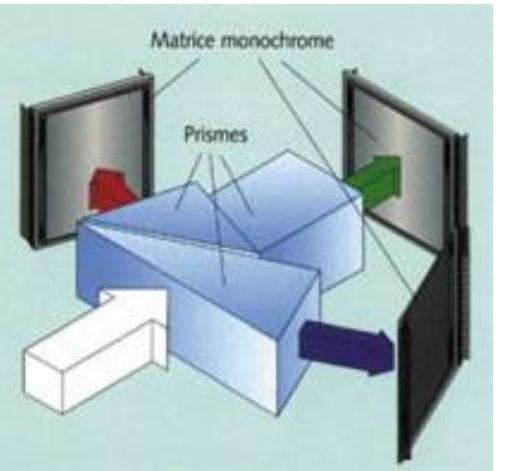
Capteur CMOS

Éléments photosensibles

Capteur Fovéon



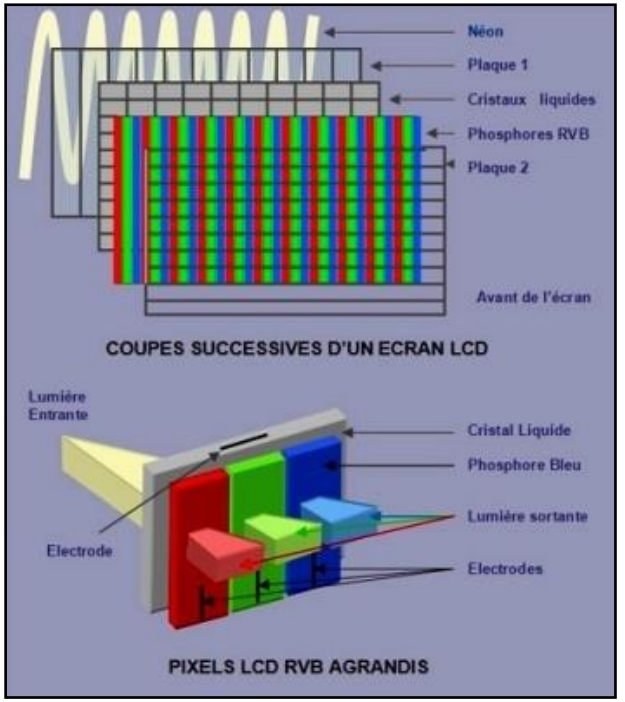
Capteur (Mono)CCD



Capteur Tri-CCD

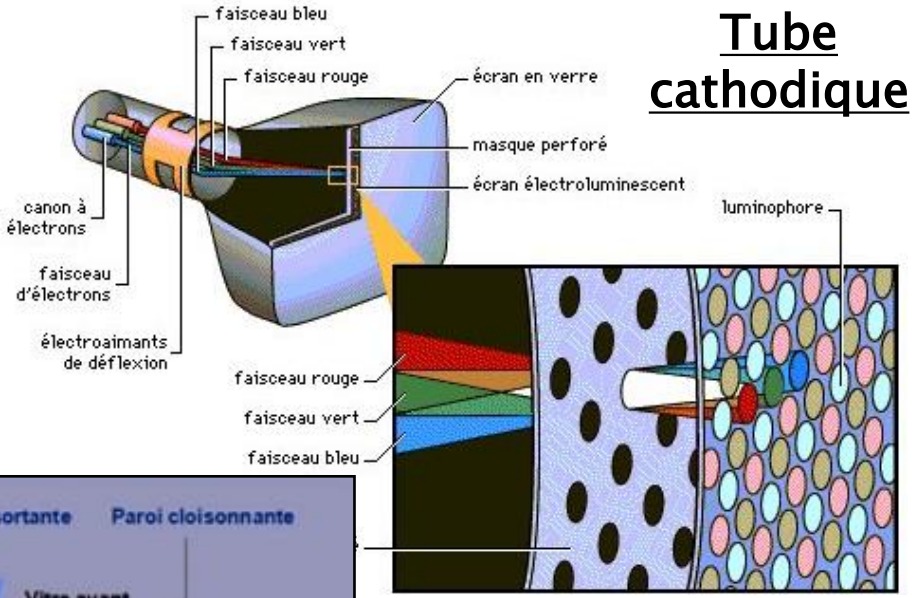
Systèmes de restitution des couleurs

Dalle LCD



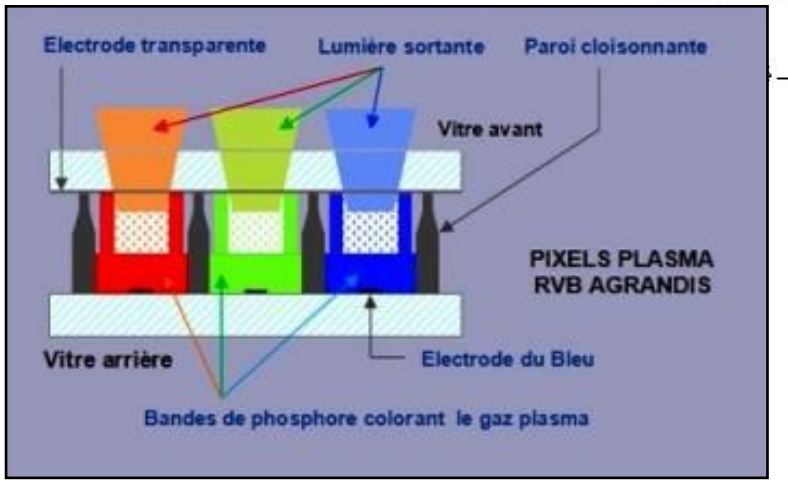
tensions électriques contrôlant l'orientation de cristaux liquides et modulant le passage d'une lumière fluorescente à travers des filtres

Tube cathodique



faisceaux d'électrons entrant en contact avec des luminophores phosphorescents

Dalle Plasma



mélanges de gaz inerte excités par des décharges électriques et transformés en plasma produisant une lumière initialement ultra-violette

primaires R, V et B dépendantes de la technologie

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 array

Grand total is 585990 elements using 585990 bytes

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

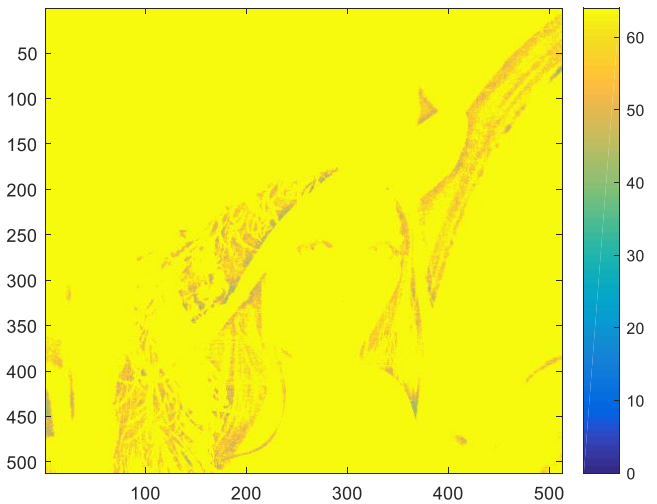
hauteur

largeur

« vraies » couleurs

Un exemple en couleurs indexées

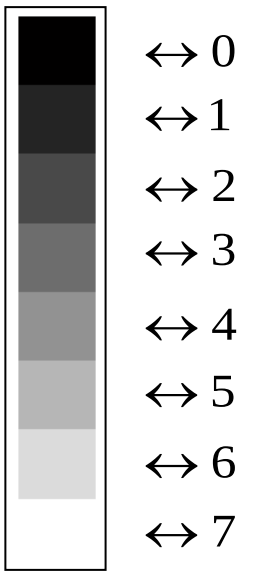
```
>> A=imread('lena.bmp');
>> whos
  Name        Size                     Bytes  Class

  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 ?

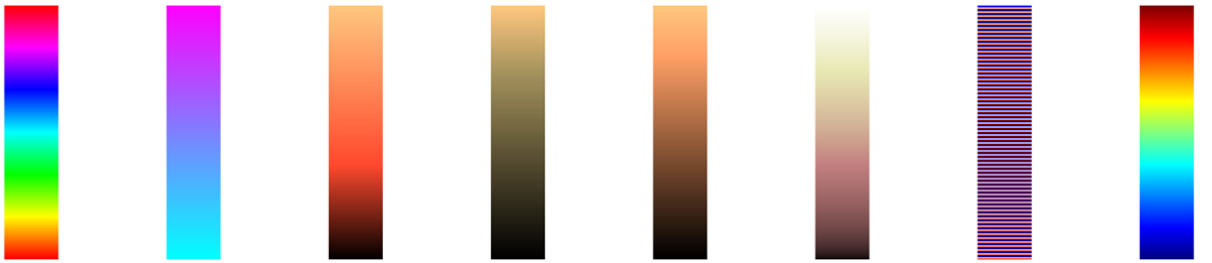
```
>> gray(8)

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



Fichier et palette

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

Fonctions Matlab de visualisation

	image	imagesc	imshow
Formats	uint8, uint16, double		uint8, uint16, single, double, …
Palette initiale	parula (64 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 (auto. par défaut/ contrôlée [.])	Toolbox « Image Processing »

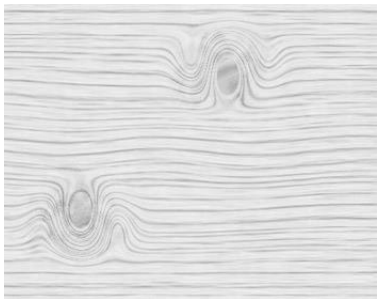
Méthodologie (1/4)

- Bilan des caractéristiques de l'image
  - Dimensions spatiales
  - Format numérique
  - Intervalles d'intensité
  - Codage
    - ✓ « vraies » couleurs
    - ✓ couleurs indexées
- Identification de la fonction d'affichage
  - Contexte
    - ✓ de « fidélité »
    - ✓ « informationnel »
    - ✓ de comparaison
  - Choix : palette, ratio L/H, intervalles d'intensité, etc.

Méthodologie (2/4)

Contexte « de fidélité »

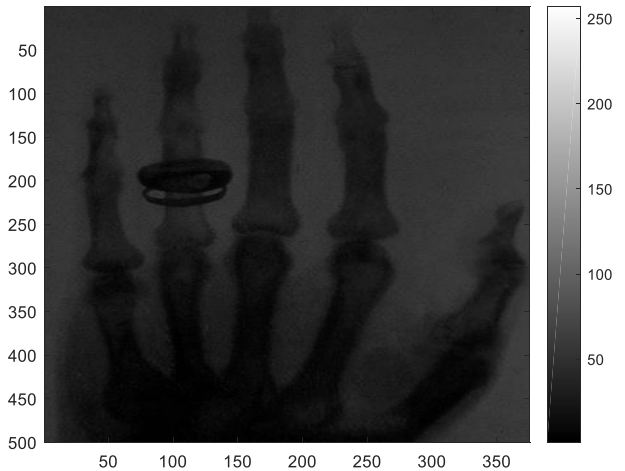
- Respect des intensités initiales
- **imshow/image**

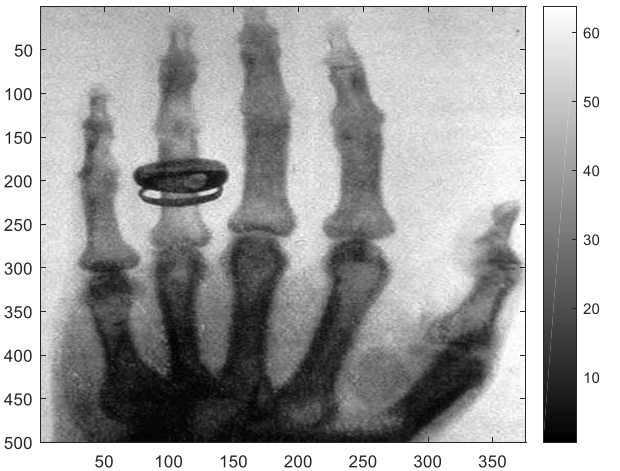


Méthodologie (3/4)

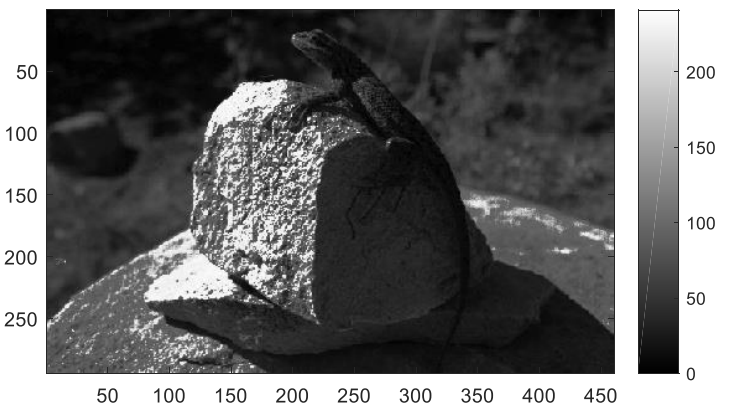
Contexte « informationnel » (couleurs indexées uniquement)

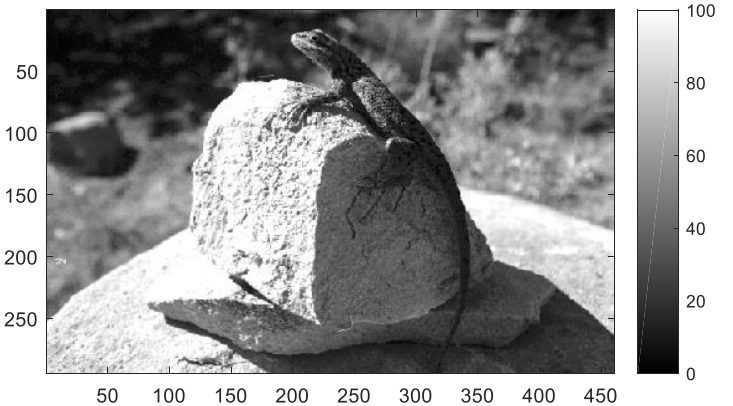
- Etalement automatique des intensités (**imagesc**)





- Etalement contrôlé des intensités (**imagesc/imshow(…,[ ])**)

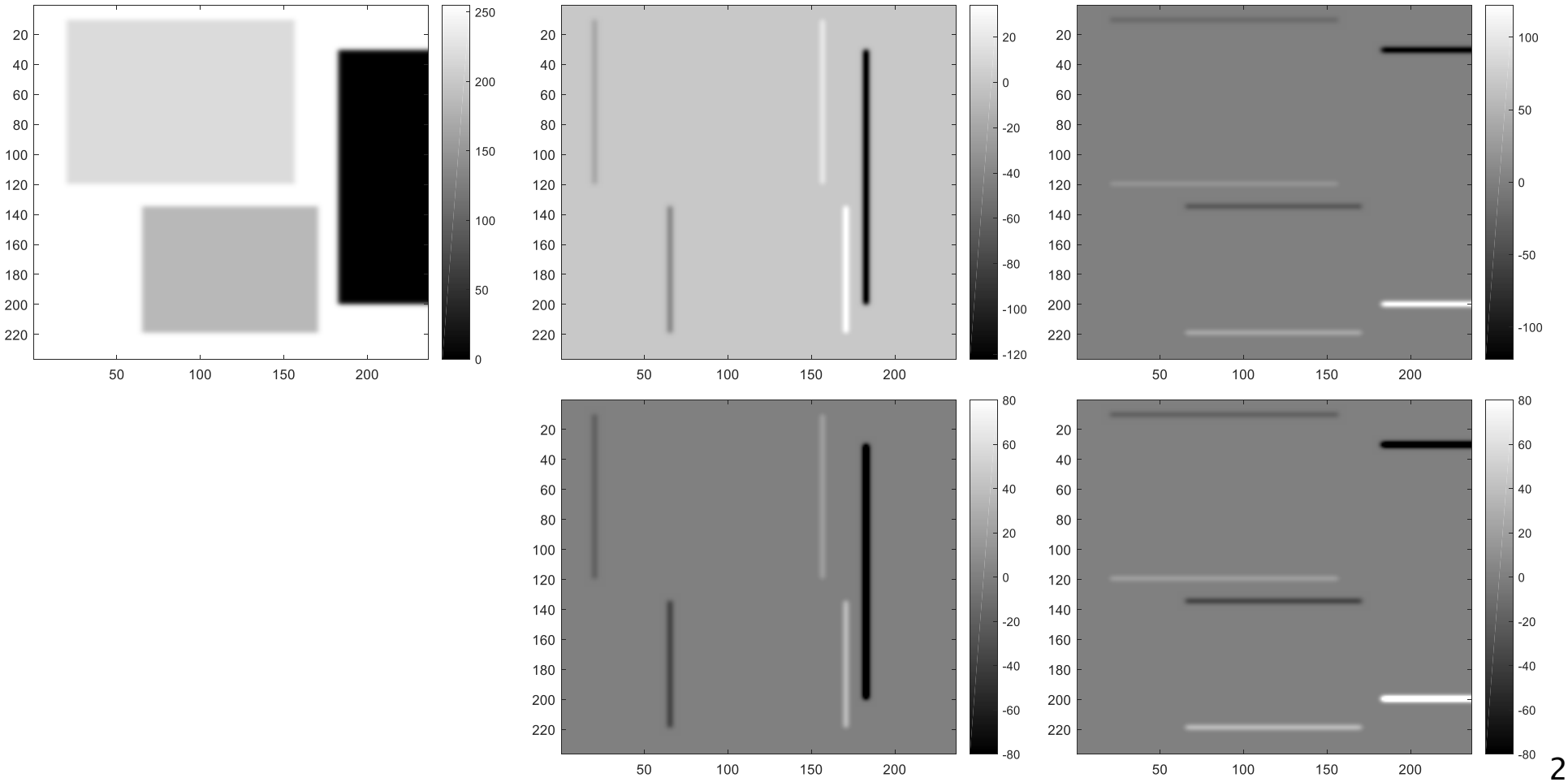




Méthodologie (4/4)

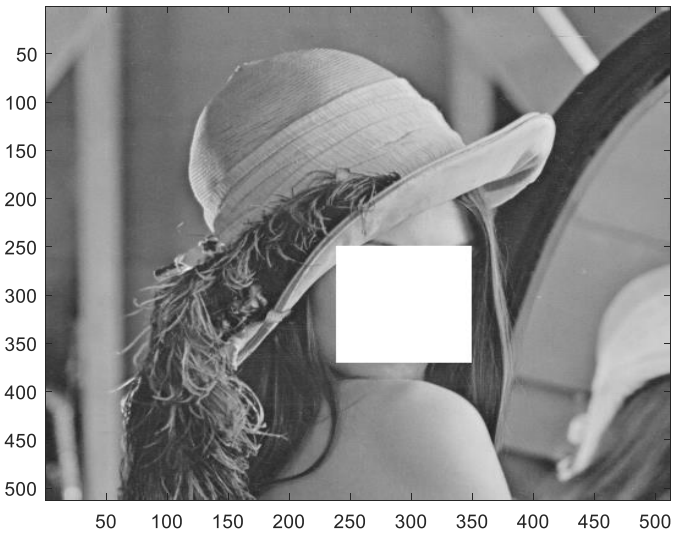
Contexte de comparaison (couleurs indexées uniquement)

- Etalement contrôlé et identique des intensités



Formats et opérations

```
clear
close all

A=imread('lena.bmp');
figure, image(A)
colormap(gray(256))

B=zeros(512);
B(250:370,240:350)=300;

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



Challenge 1 : format numérique

```
clear
close all

s=load('challenge.mat');
I=s.A;
```

Challenge 2 : ratio L/H

```
clear
close all

s=load('challenge.mat');
I=s.B;
```

Challenge 3 : intervalles d'intensité

```
clear
close all

s=load('challenge.mat');
I=s.C;
```

Challenge 4 : palette continue

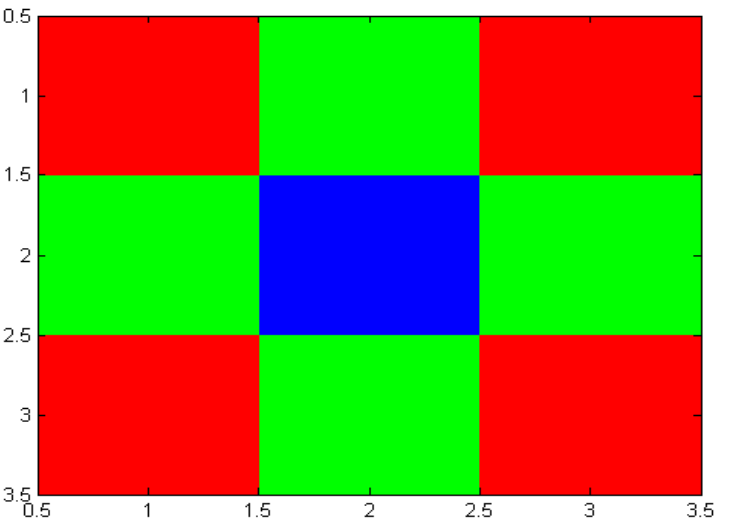
```
clear
close all

s=load('challenge.mat');
I=s.D;
```

Challenge 5 : palette « discrète »

```
clear
close all

s=load('challenge.mat');
I=s.E;
```

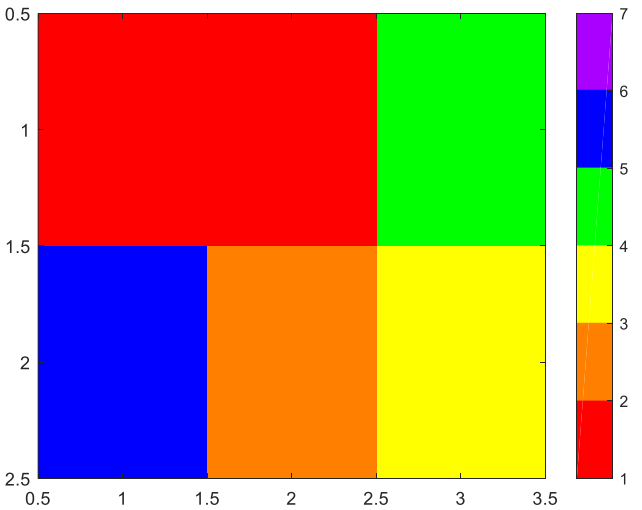
Synthèse en « vraies couleurs »

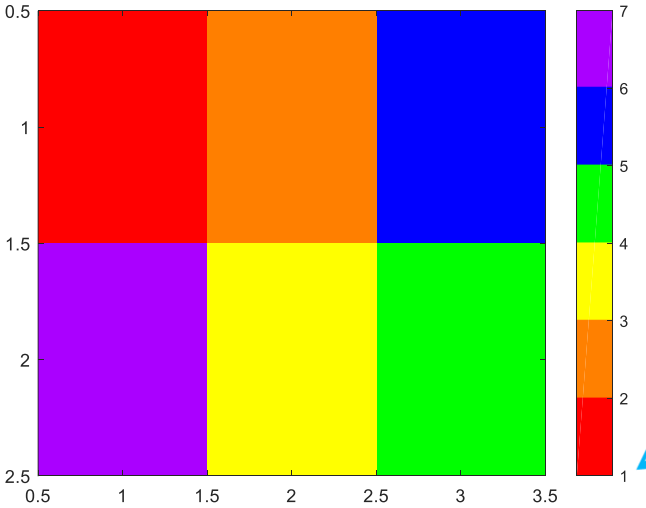
```
clear
close all

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)
```

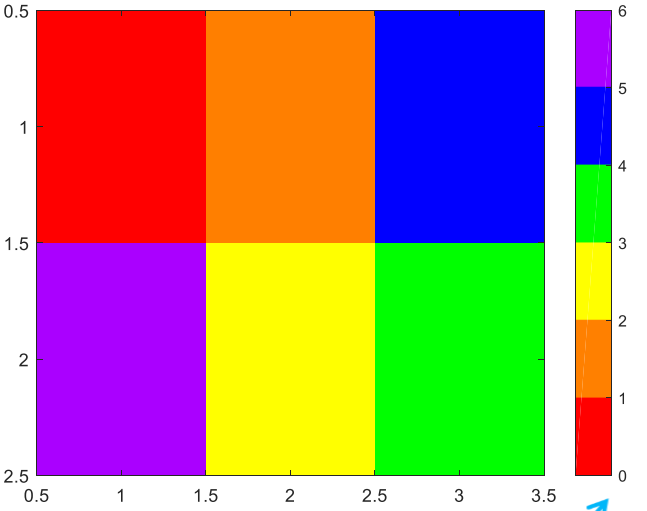


Synthèse en couleurs indexées

```
clear, close all

img=[0 1 4;5 2 3];
figure
image(img)
colormap(prism(6))
colorbar
figure
image(uint8(img))
colormap(prism(6))
colorbar
figure,
image(img+1)
colormap(prism(6))
colorbar
```



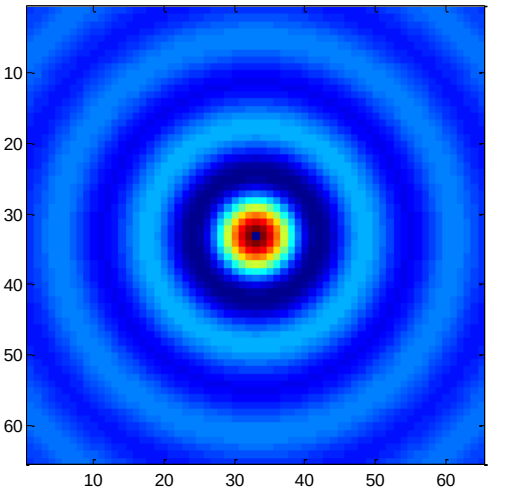


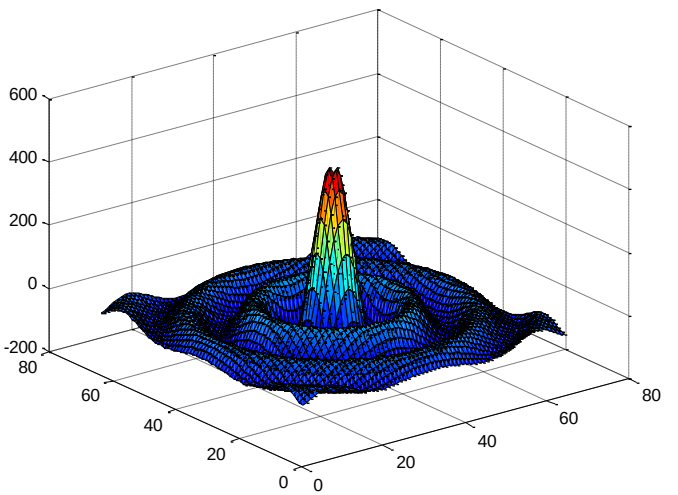


0 ou 1 !!!

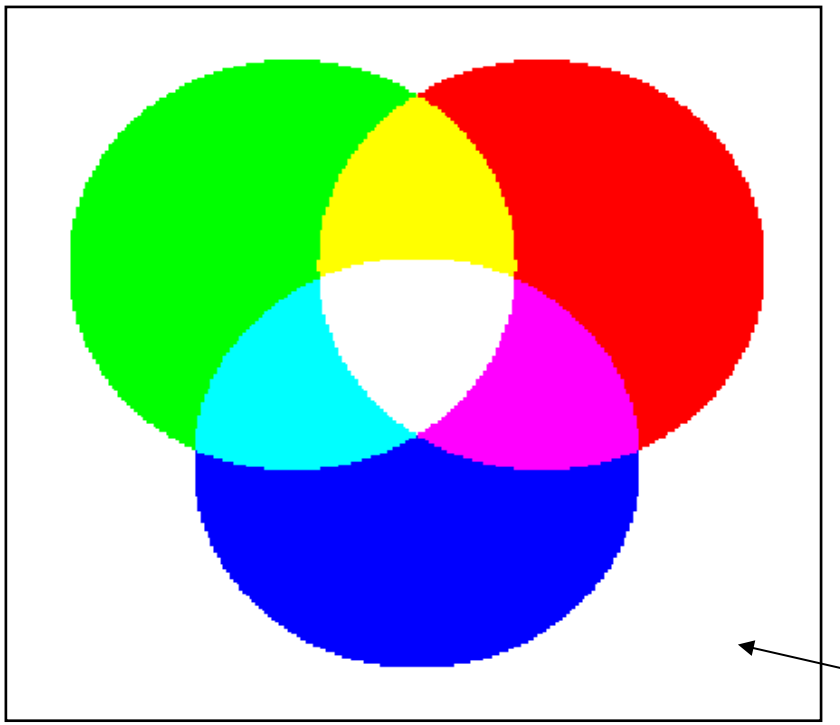
Synthèse analytique : meshgrid

```
clear
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

« 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=1:100
    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
  - 100 images
  - 5 images/seconde
  - sans compression

Exercice : interaction

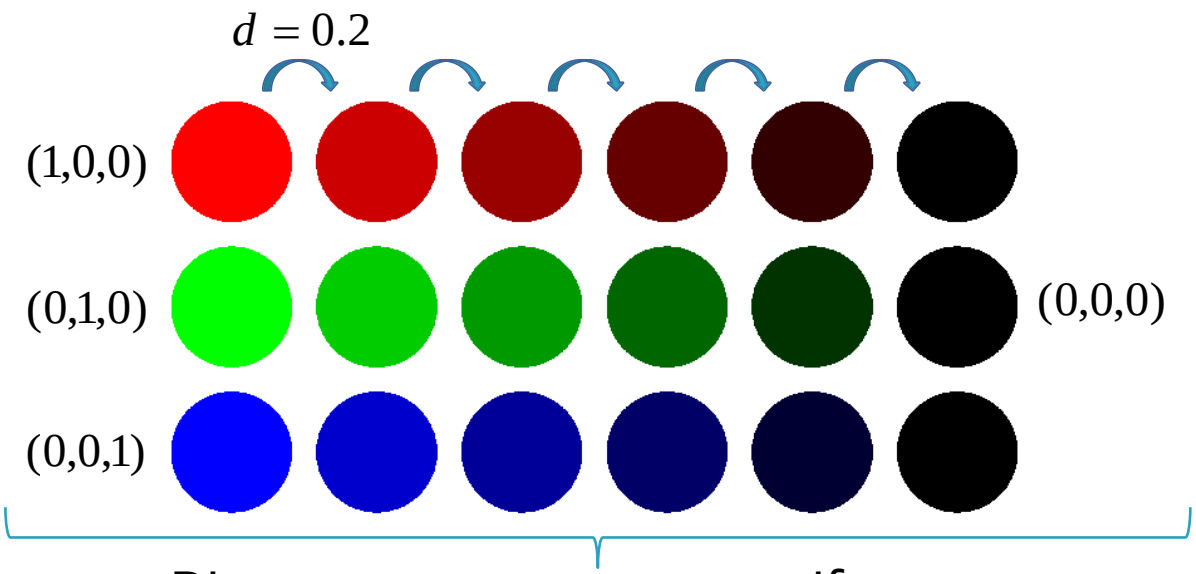
- Affichage d'une image
- Capture d'un segment (ginput), de gauche à droite et de haut en bas
- Extraction du profil (signal horizontal ou vertical le plus « proche ») correspondant
- Affichage du profil

Initiation au Traitement des Images

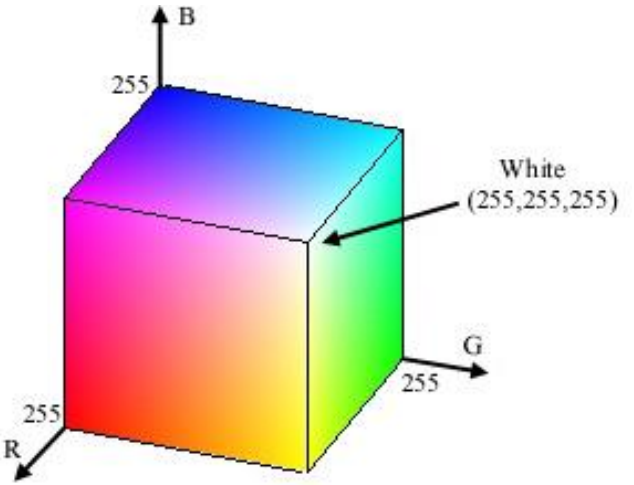
- Couleur/Amplitude
- Spatial
- Fréquentiel

Espaces caractéristiques

Espace chromatique RVB(RGB)



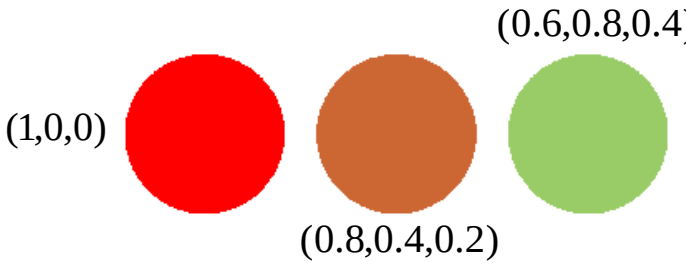
Distances perçues non « uniformes »



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

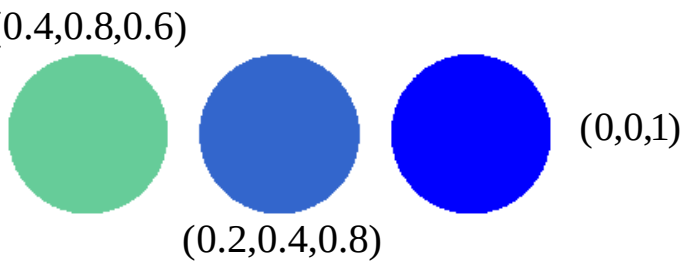
luminance (intensité moyenne)

Espace d'acquisition du système visuel humain

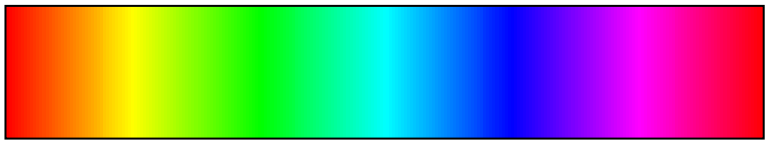


R, G, B ?

décomposition très difficile (même à 20% près)



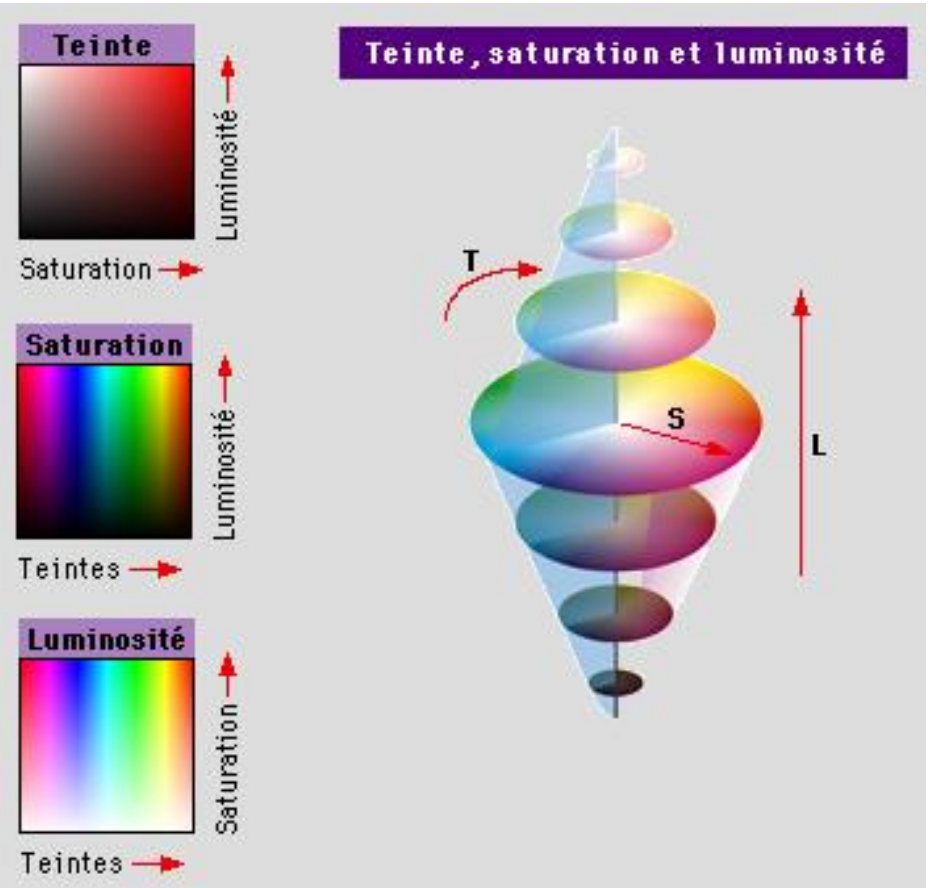
Espace chromatique TSL (HSL)



Teinte

Saturation ← « mélange » avec le gris

Luminosité ← « mélange » avec le noir



Espace de description naturelle

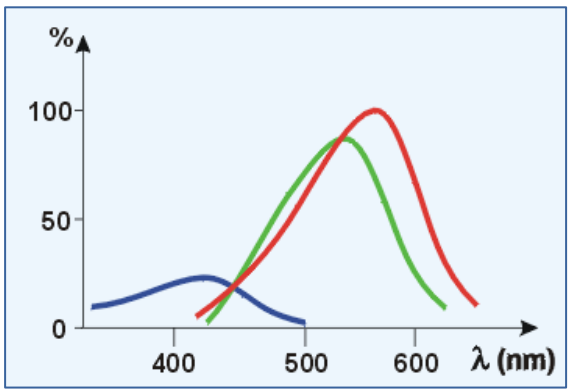
Luminance vs Chrominance

Niveaux de gris

luminosité plus forte

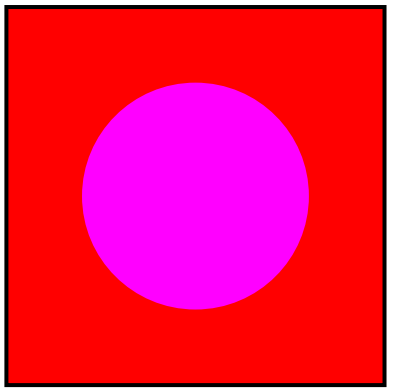
contrastes plus faibles

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

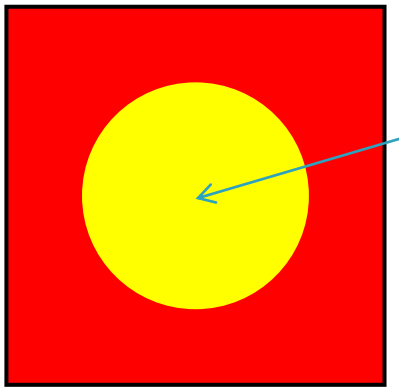


Sensibilité relative des photo-récepteurs de chrominance

Luminance « perceptuelle »

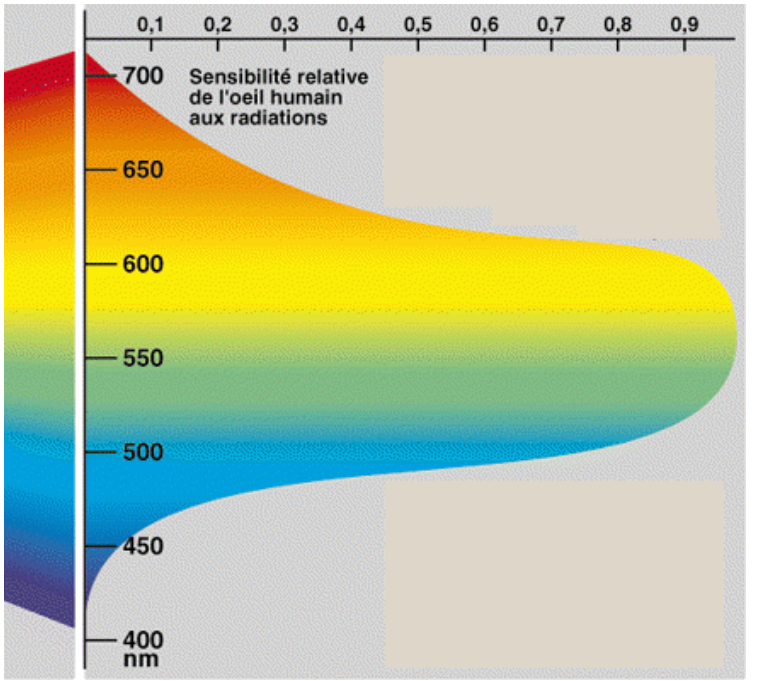


Rouge/Bleu



luminosité plus forte

Rouge/Vert



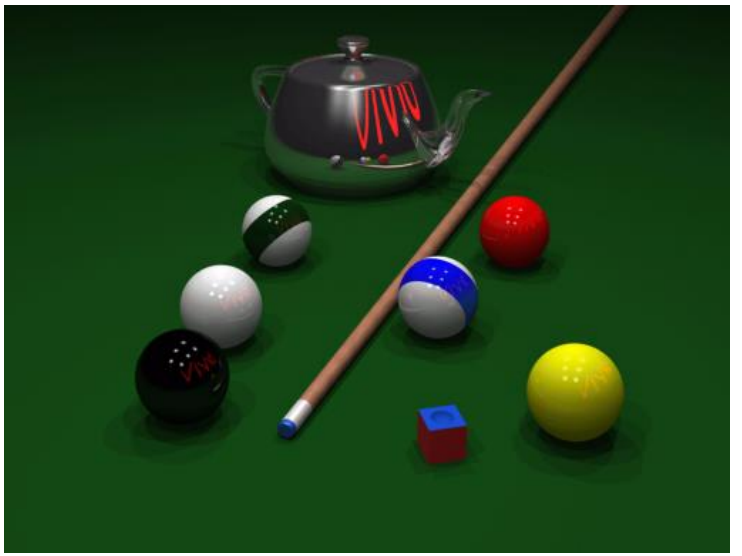
$$Y = 0.299\ R + 0.587\ V + 0.114\ B$$

pondération plus faible

Espace colorimétrique YCbCr

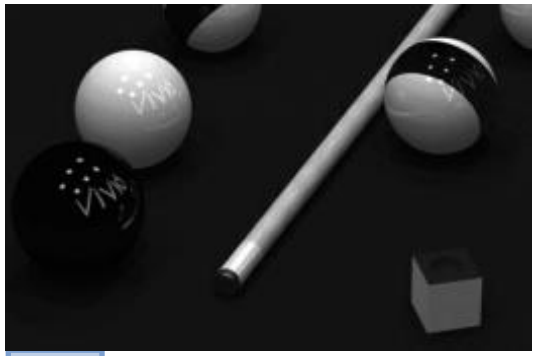
```
clear
close all

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,:)))
```

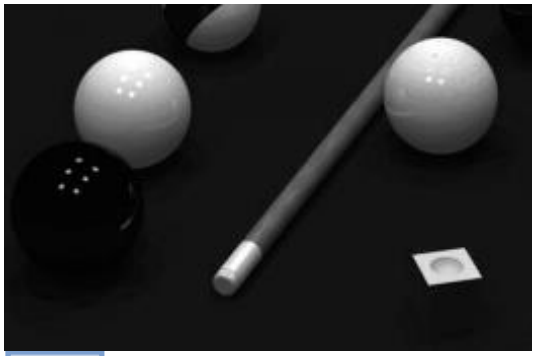



L

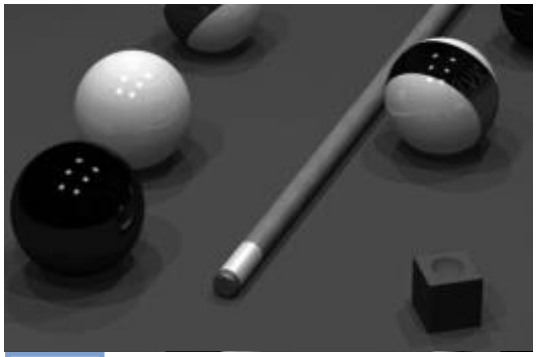

Y



R

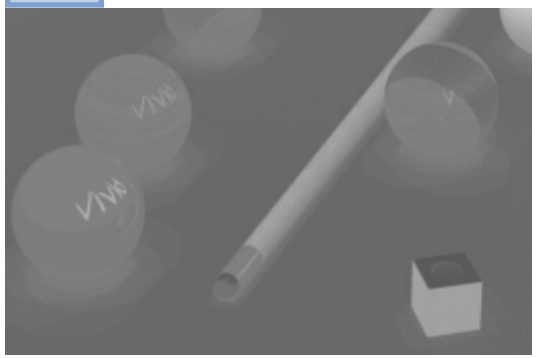


B

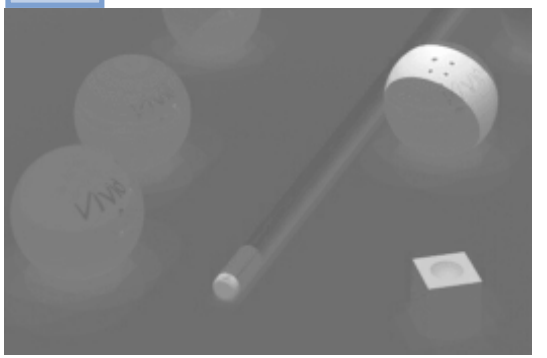


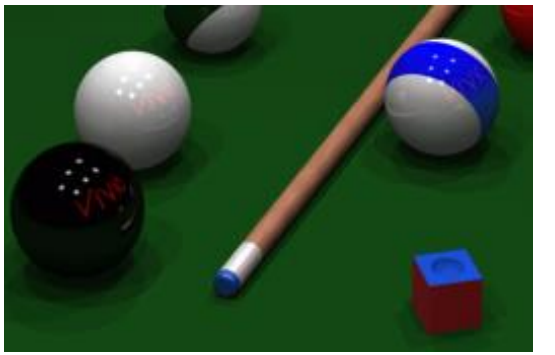
G

C_r

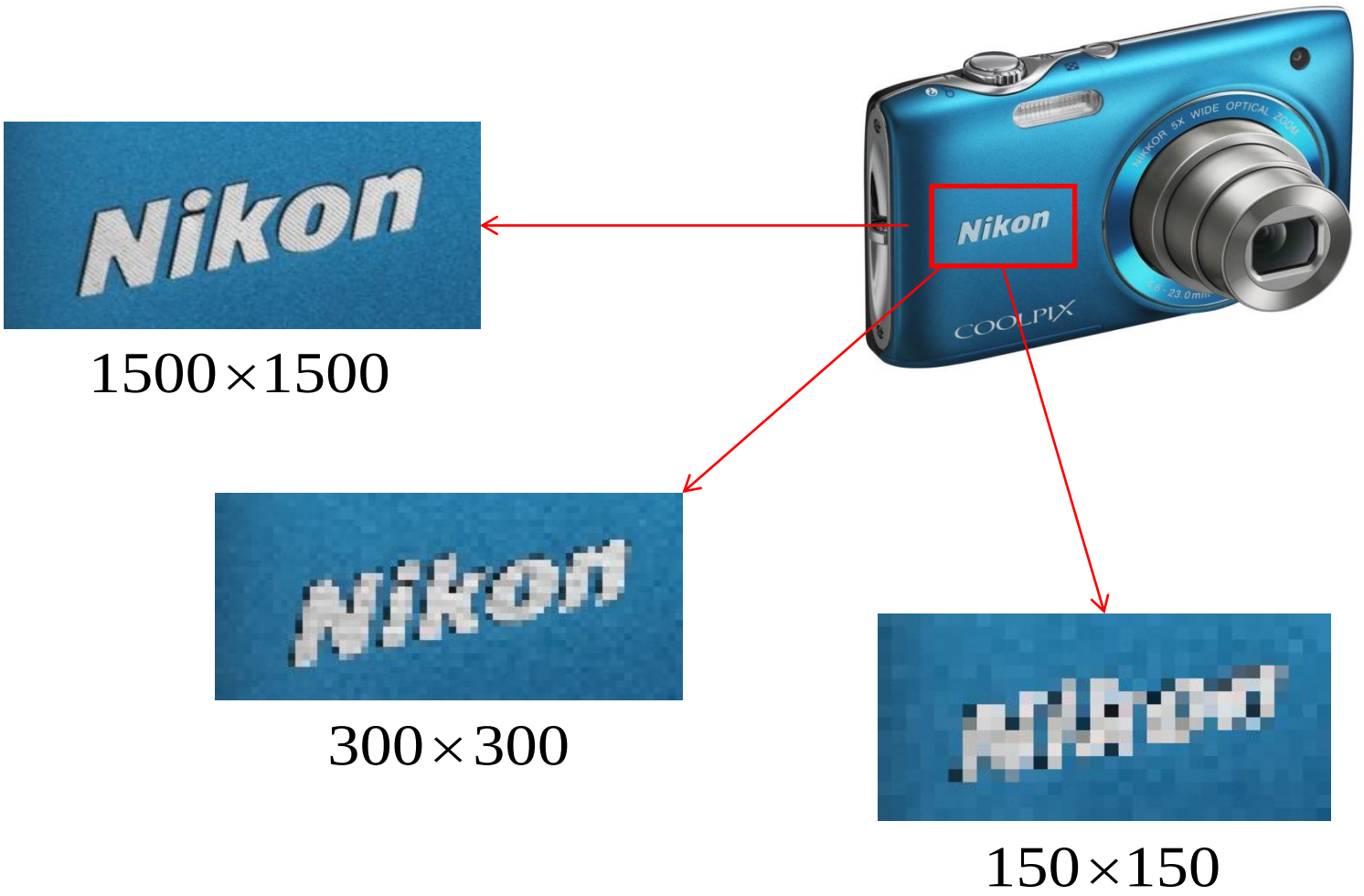


C_b

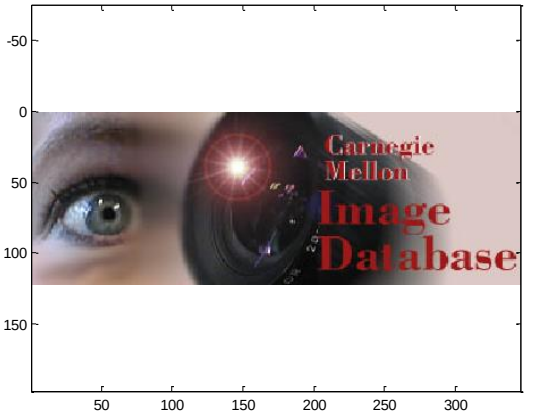


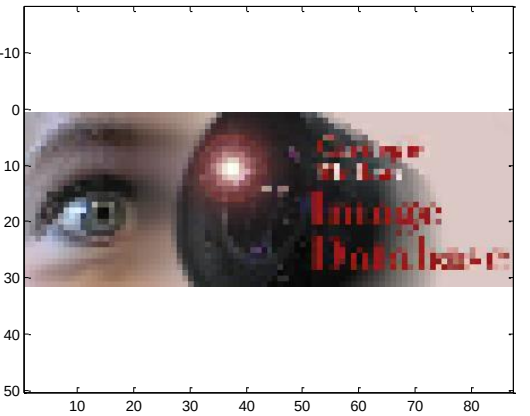


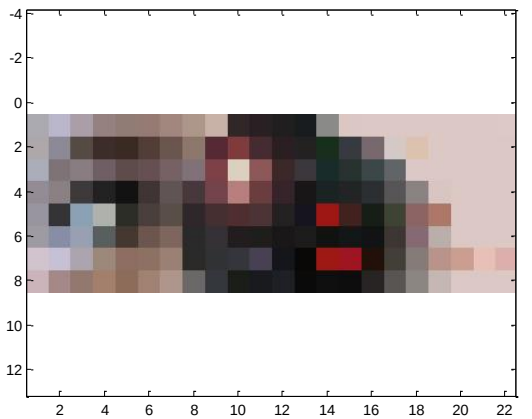
Résolution spatiale



```
clear
close all

A=imread('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')
```

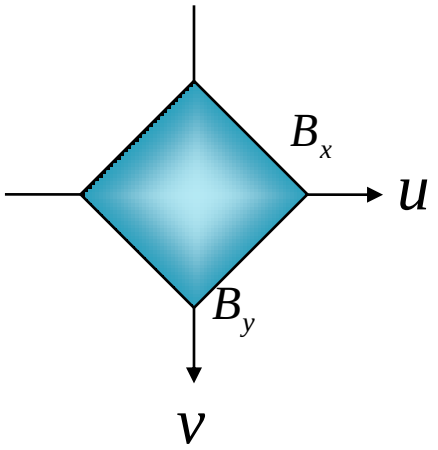




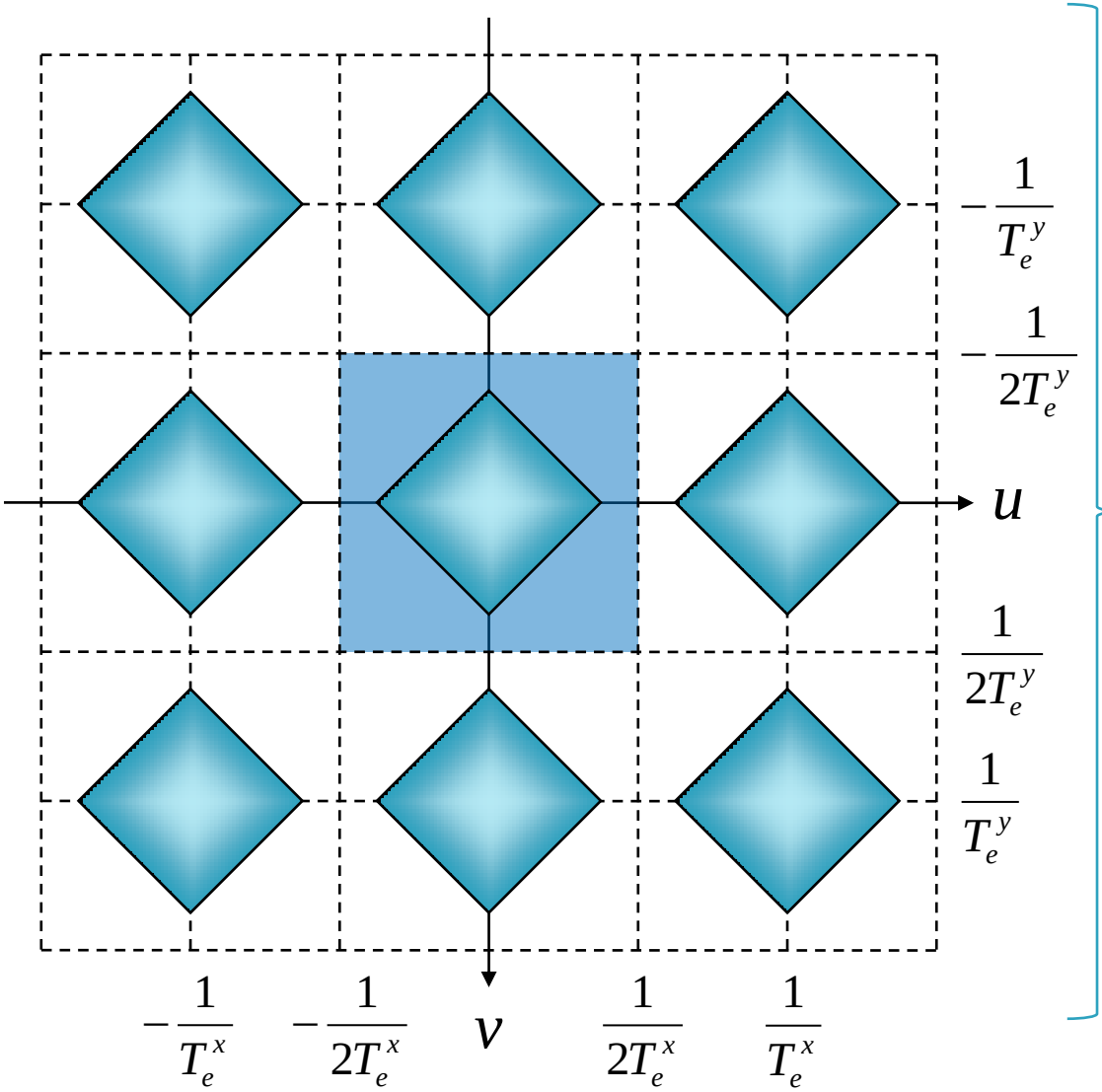


Spectre d'une séquence discrète 2d

$|F_a(u)|$



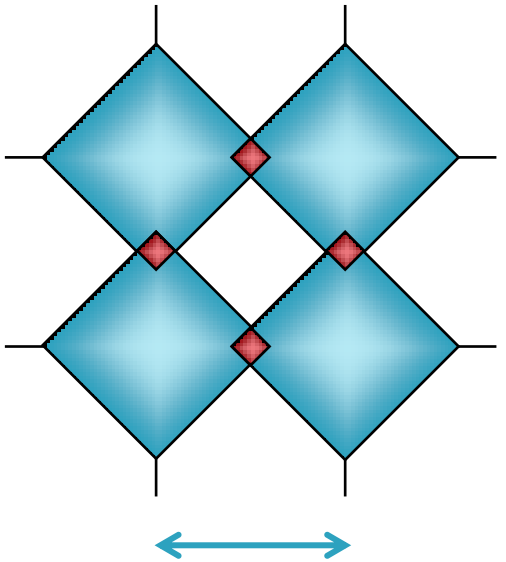
Signal 2d à bande passante limitée



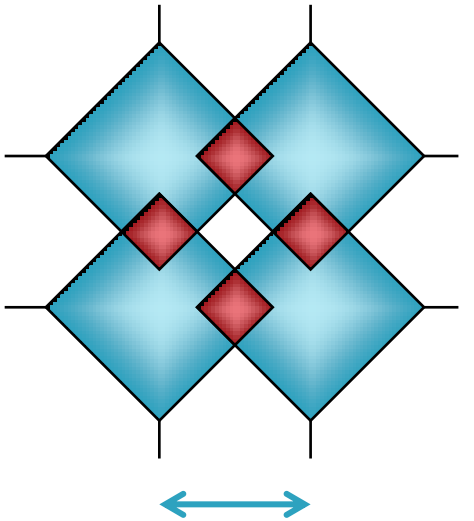
périodisation du spectre

$|F_n(u)|$

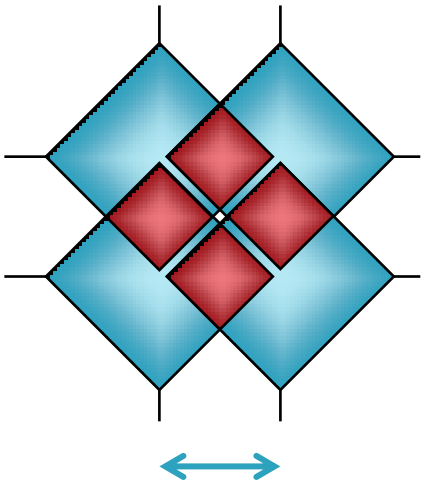
Recouvrement fréquentiel



$$\frac{1}{T_e^x}$$



$$\frac{1}{T_e^x}$$



$$\frac{1}{T_e^x}$$

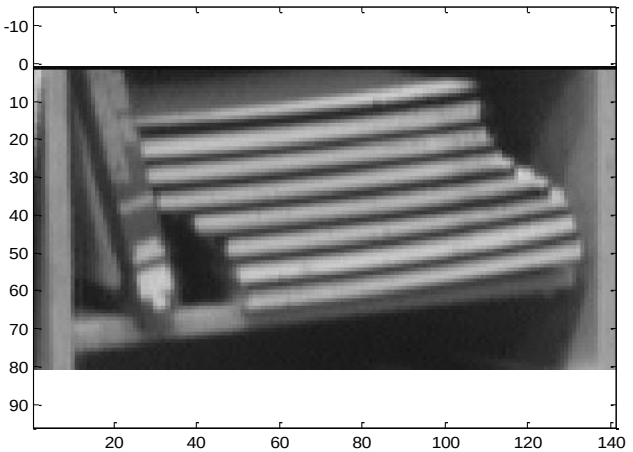
Des hautes fréquences … aux basses fréquences

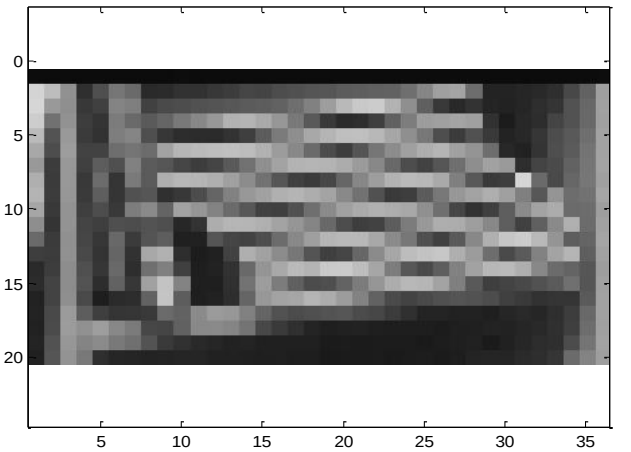
```
clear
close all

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

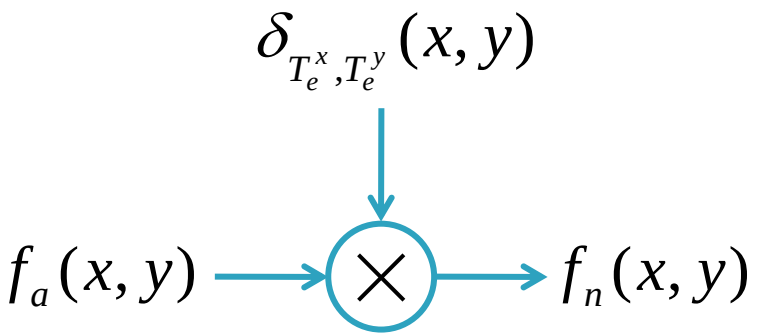
Extension au cas des signaux bidimensionnels (images)

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

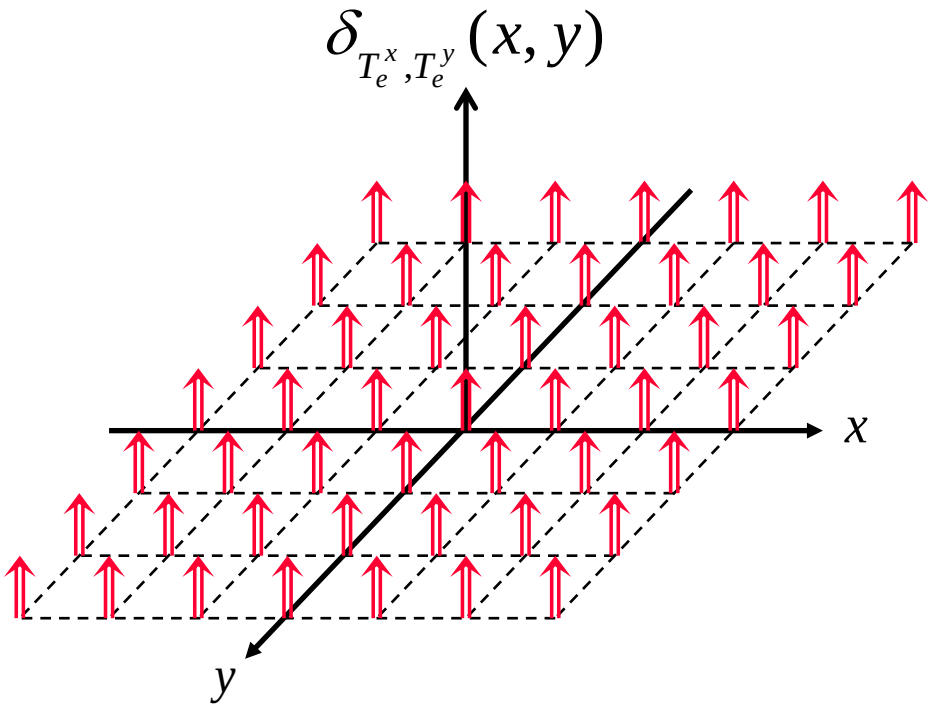
$$f_n(x,y) = \begin{cases} f_a(x,y) & \text{pour} \quad \begin{cases} x = mT_e^x \\ y = nT_e^y \end{cases} \\ 0 & \text{sinon} \end{cases}$$



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

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

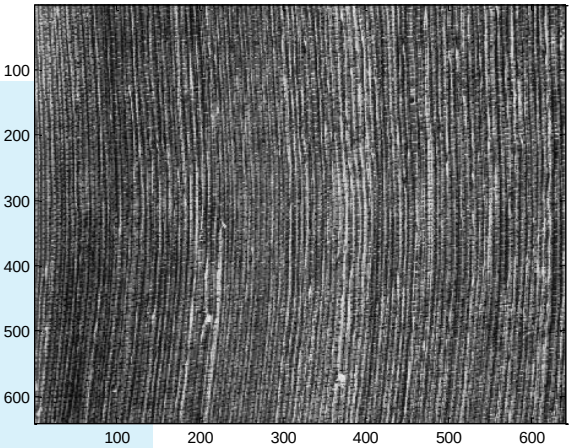
Transformée de Fourier discrète (d'une séquence discrète) (TFD)

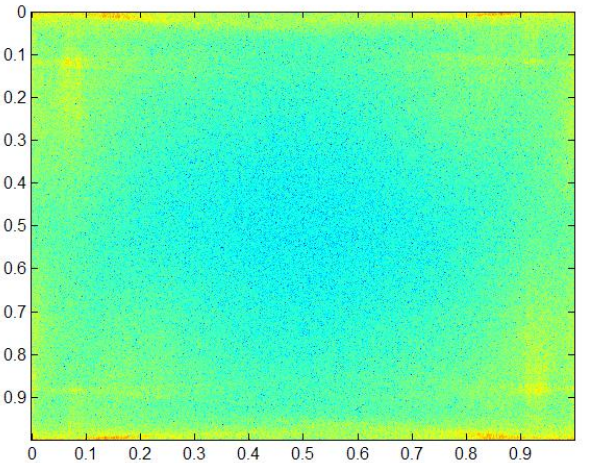
$$F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi\left(k\frac{m}{M}+l\frac{n}{N}\right)}$$

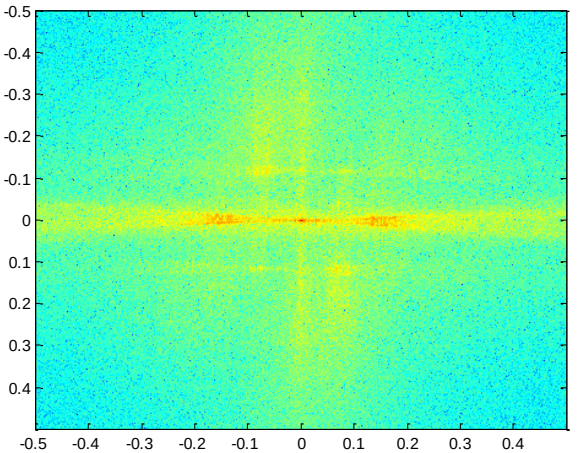
variables discrètes $\begin{cases} k \in [0, M-1] \\ l \in [0, N-1] \end{cases}$

TFD

```
clear
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));
```

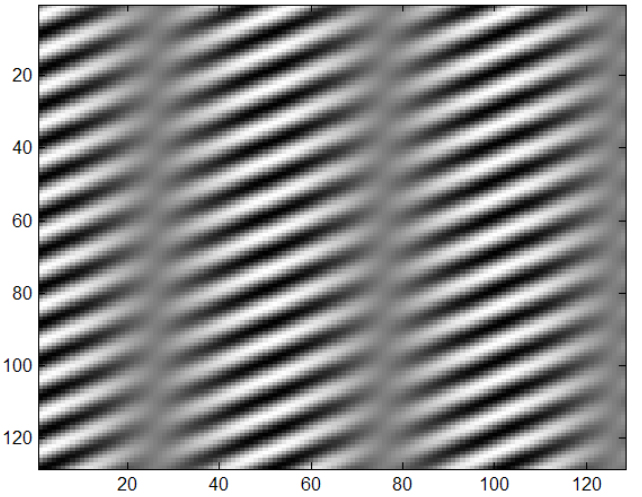


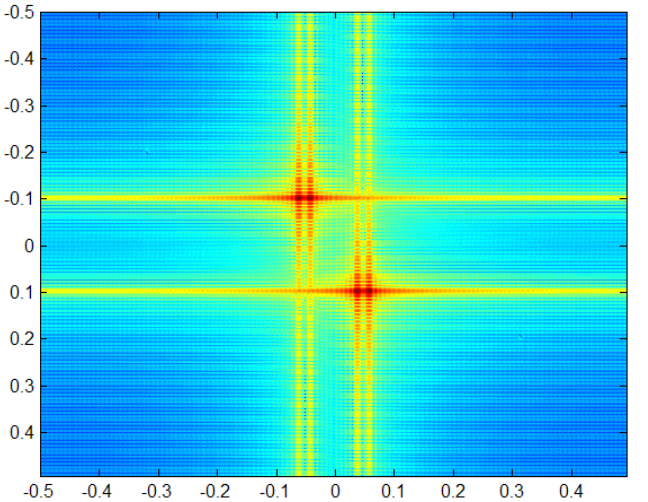




TFD et fenêtrage

```
clear
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)
```



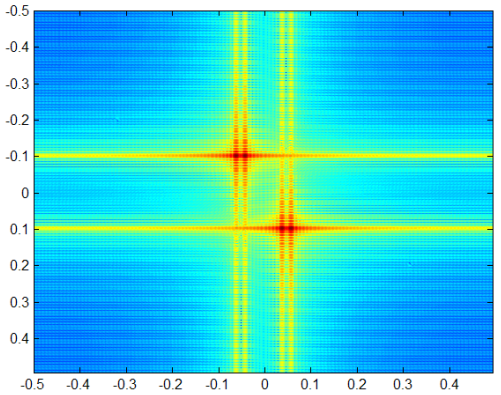


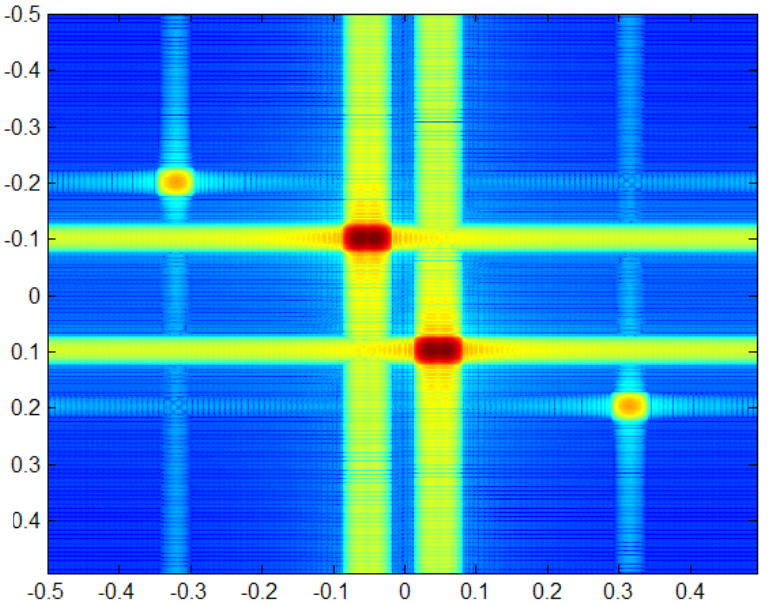
Kaiser séparable

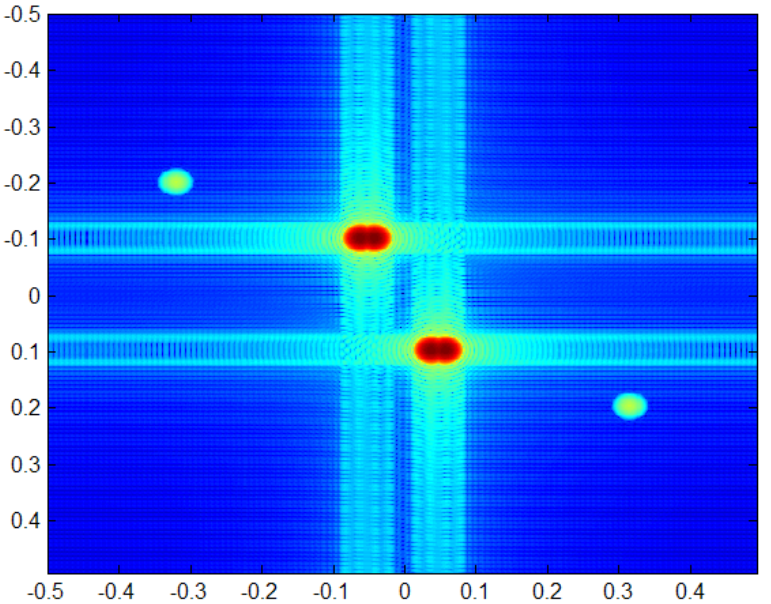
```
W=kaiser(128,10)*kaiser(128,10)';
Iw=I.*W;
If2=fftshift(log10(abs(fft2(Iw,512,512))));
figure, imagesc(f,f,If2)
```

```
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

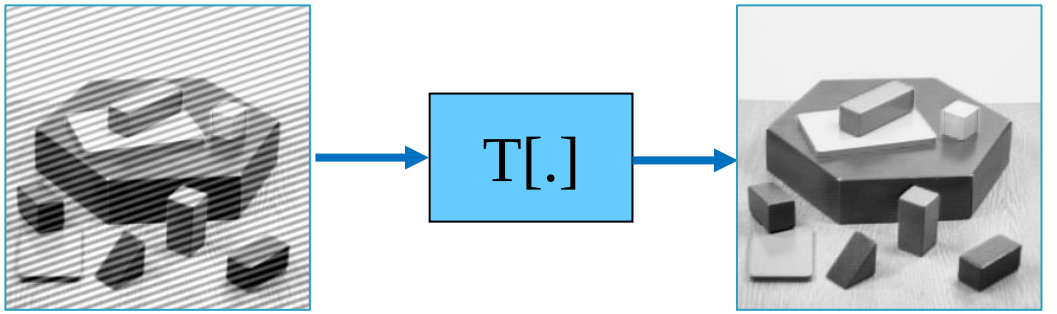

-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4

Initiation au Traitement des Images

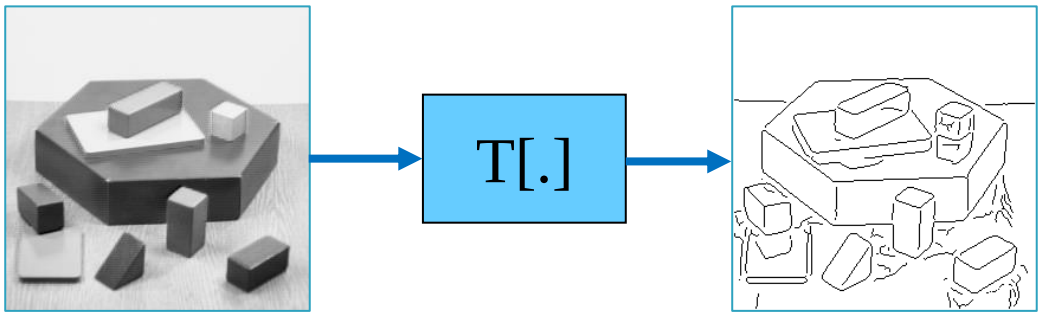
- Filtrage linéaire
  - Filtres RIF
  - Détection de contours
- Filtrage non linéaire

Traitements

Filtres RIF



Filtrage fréquentiel

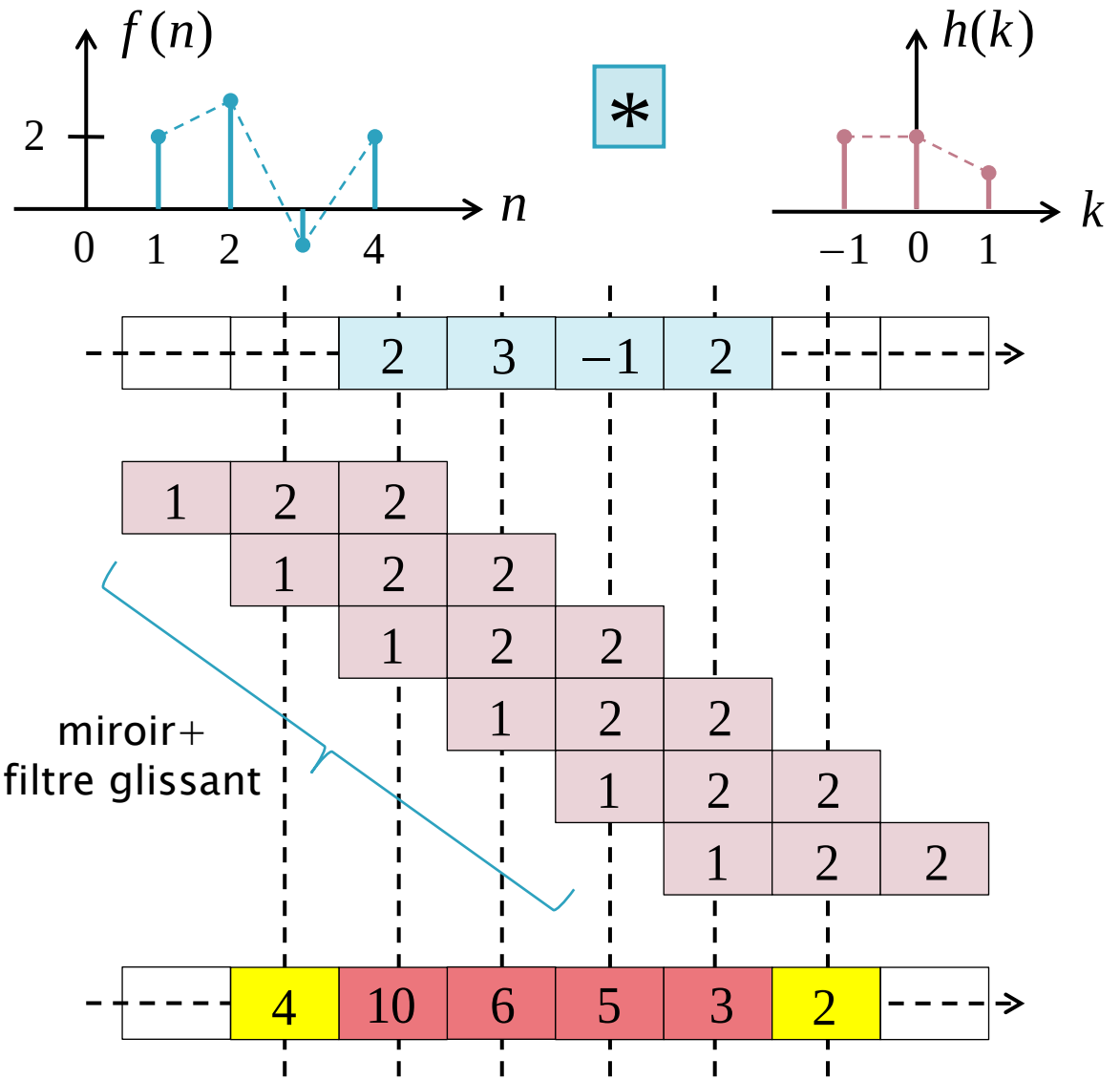


Détection de contours

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

Convolution 2D

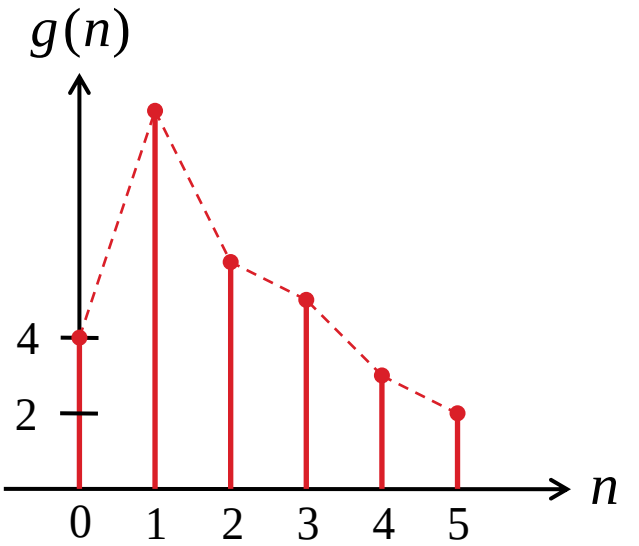
Convolution discrète 1d



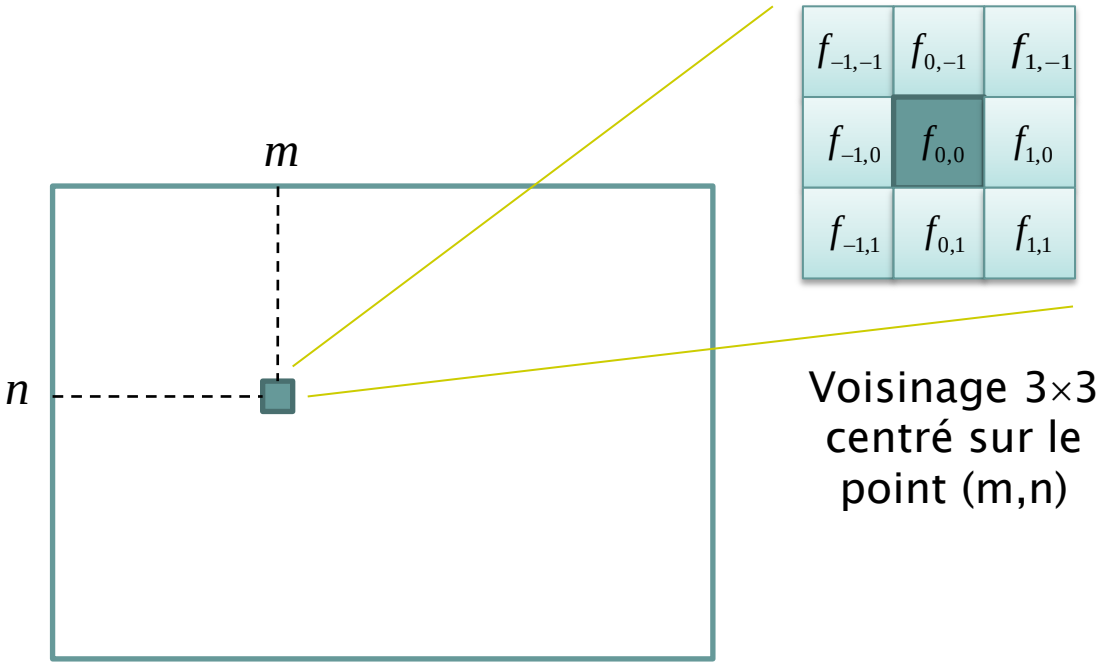
$$g(n) = \sum_{k=-K}^{K} f(n-k)h(k)$$

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

miroir



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

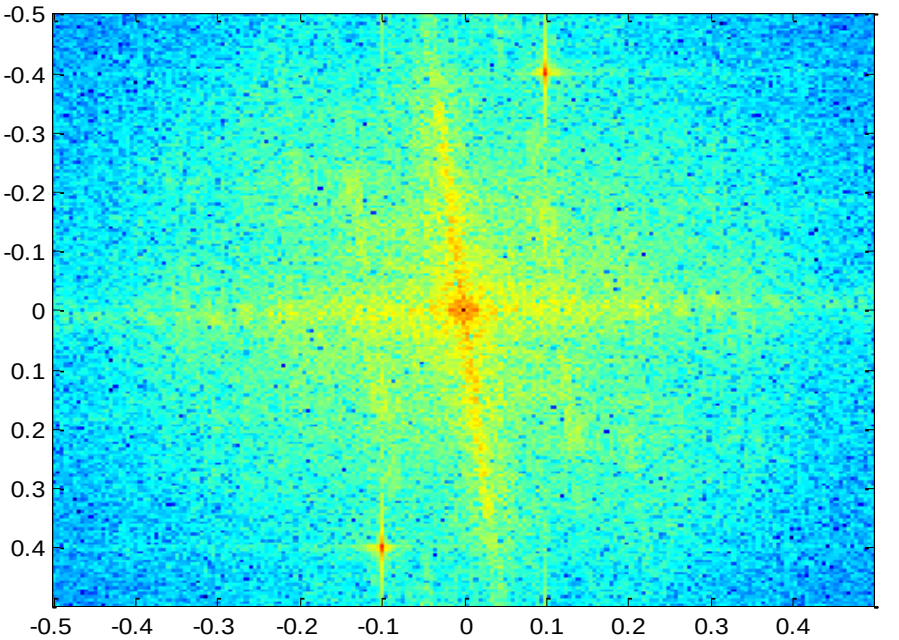
```
clear
close all

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')
```

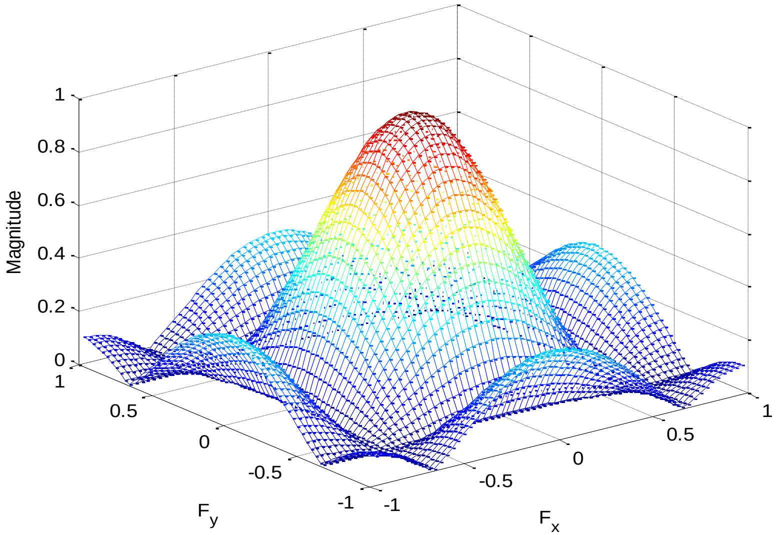
```
I =

     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
```

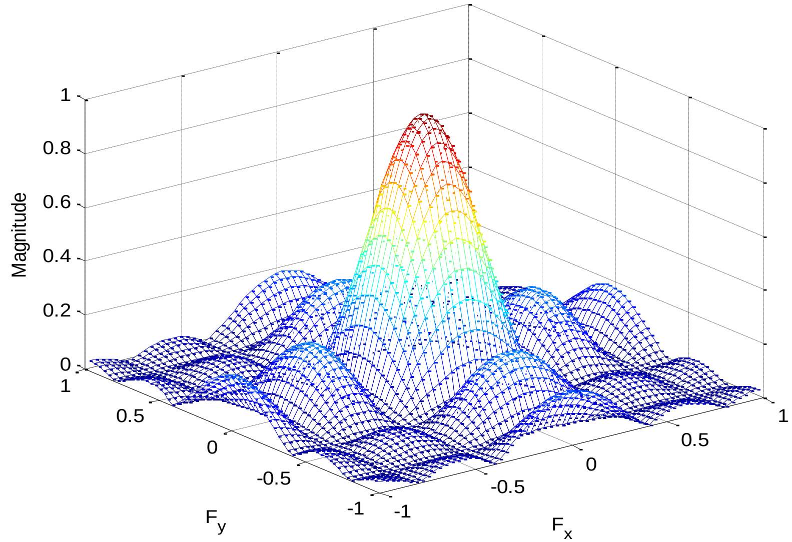
Exercice

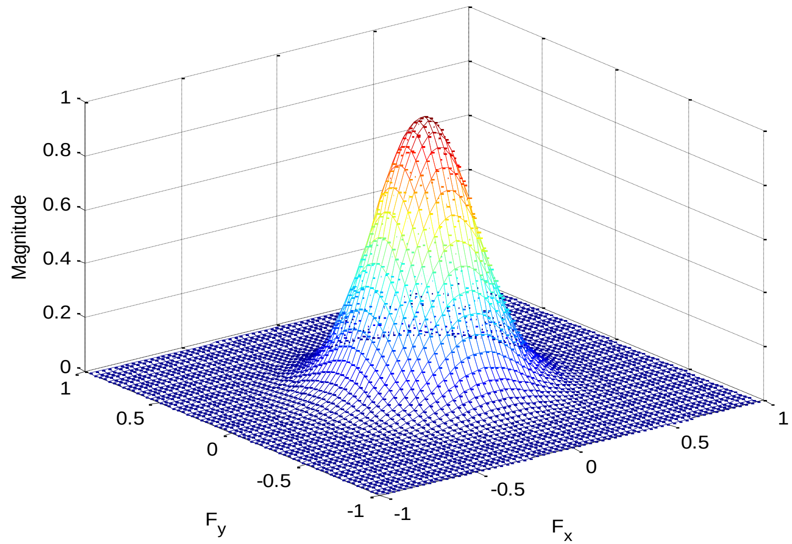
- Affichage de l'image 'monument_ext.bmp'
- Filtrage passe-bas pour éliminer/réduire le bruit
  - Justification du choix (visualisation de la TFD)
  - Vérification de l'adéquation des filtres (freqz2)
- Affichage des images filtrées et de leur spectre
- Calcul et affichage du « bruit » (différence entre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

```
clear, close all

A=double(imread('monument_ext.bmp'));
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(3)/9;
H2=ones(5)/25;
[X,Y]=meshgrid(-5:5);
sigma=1.5;
H3=exp(-(X.^2+Y.^2)/(2*sigma^2))/(2*pi*sigma*sigma);
figure, freqz2(H1)
figure, freqz2(H2)
figure, freqz2(H3)
```

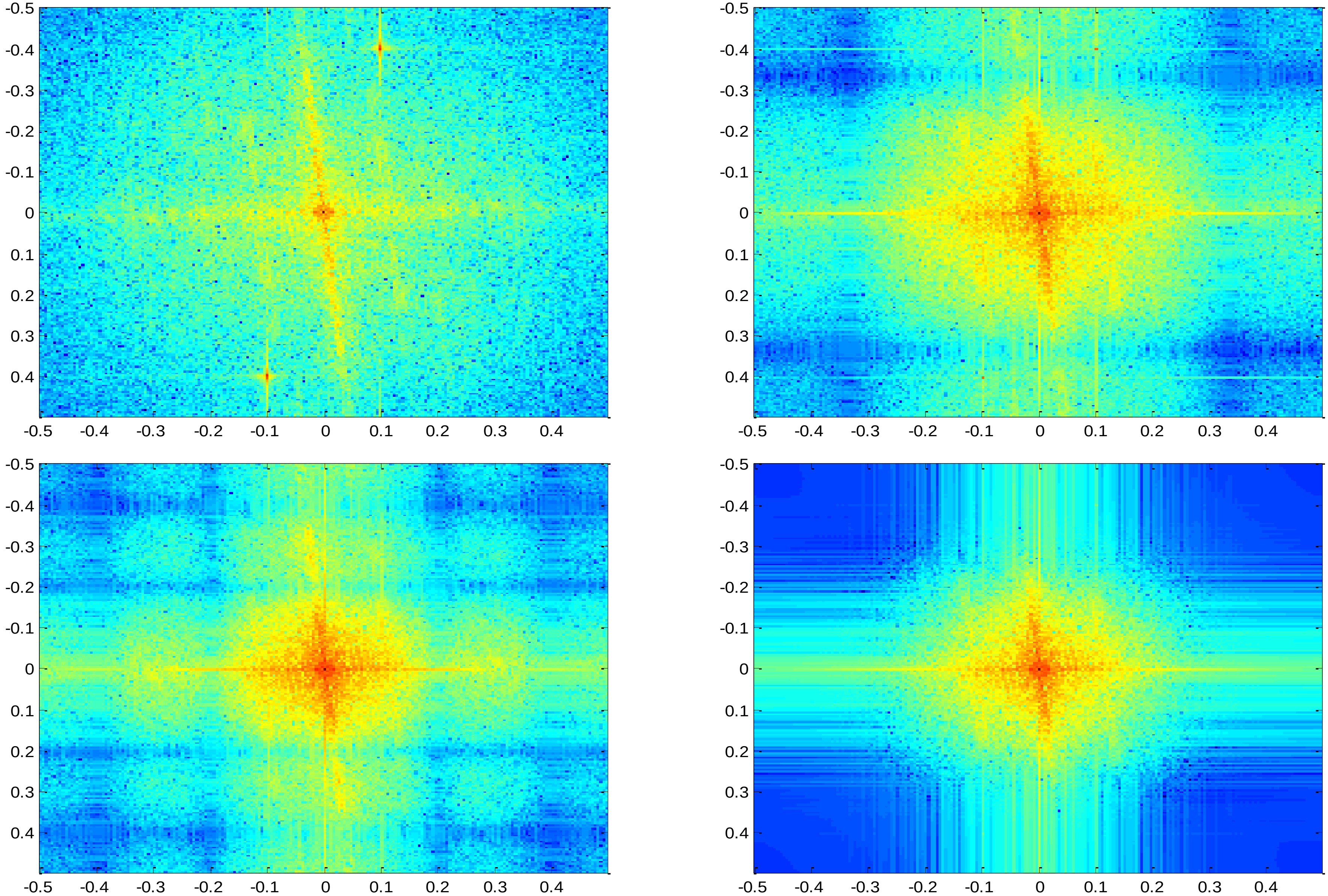

-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4


Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
Fy
-1
-0.5
0
0.5
1
Fx

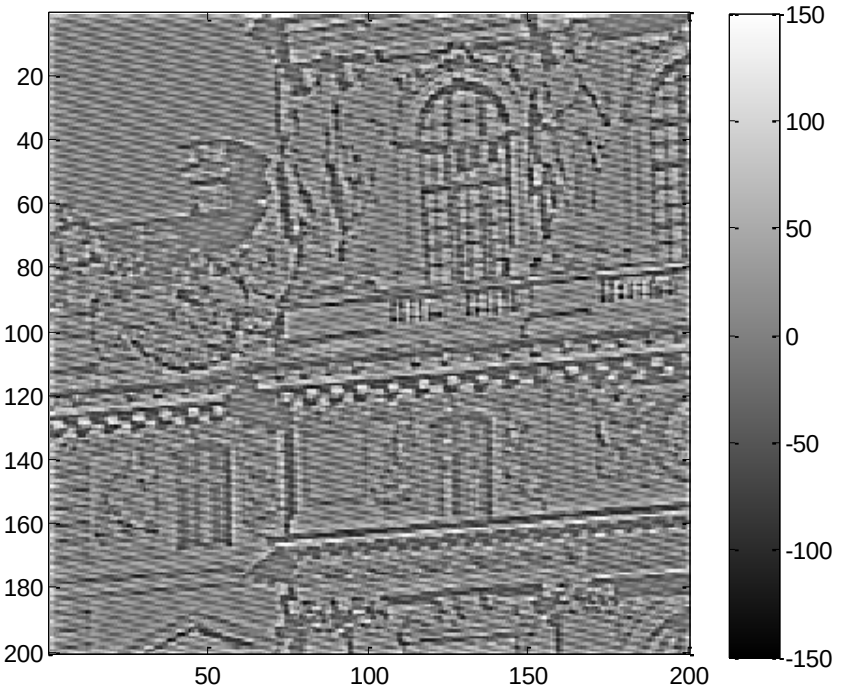

Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
Fy
-1
-0.5
0
0.5
1
Fx


Magnitude
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
Fy
-1
-0.5
0
0.5
1
Fx

```
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));

IfB=fftshift(log10(abs(fft2(B))));
figure, imagesc(fx,fy,IfB)
IfC=fftshift(log10(abs(fft2(C))));
figure, imagesc(fx,fy,IfC)
IfD=fftshift(log10(abs(fft2(D))));
figure, imagesc(fx,fy,IfD)
```


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



Détection de contours

```
clear
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))/(2*pi*sigma^4);
Hy=-Y.*exp(-(X.^2+Y.^2)/(2*sigma^2))/(2*pi*sigma^4);
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)))
```

$\sigma = 0{,}75$

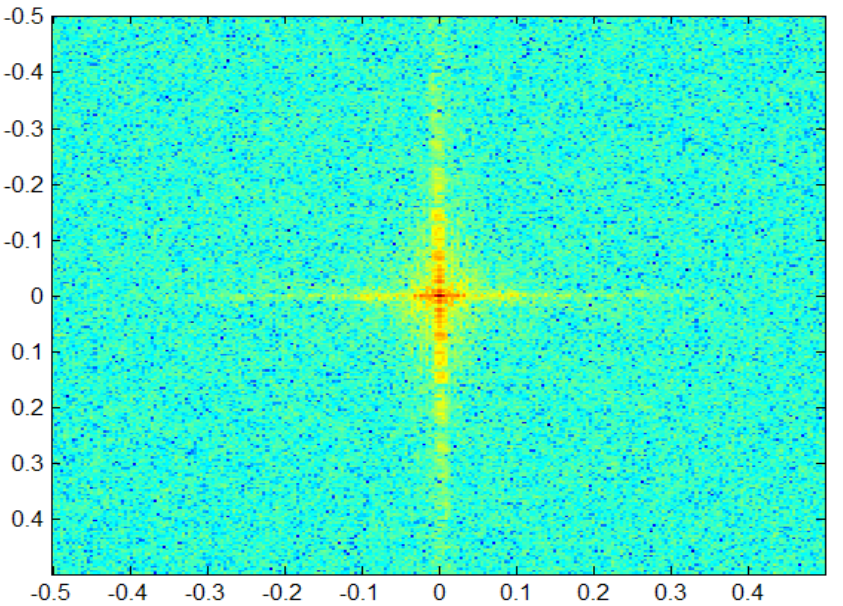
$\sigma = 1{,}5$

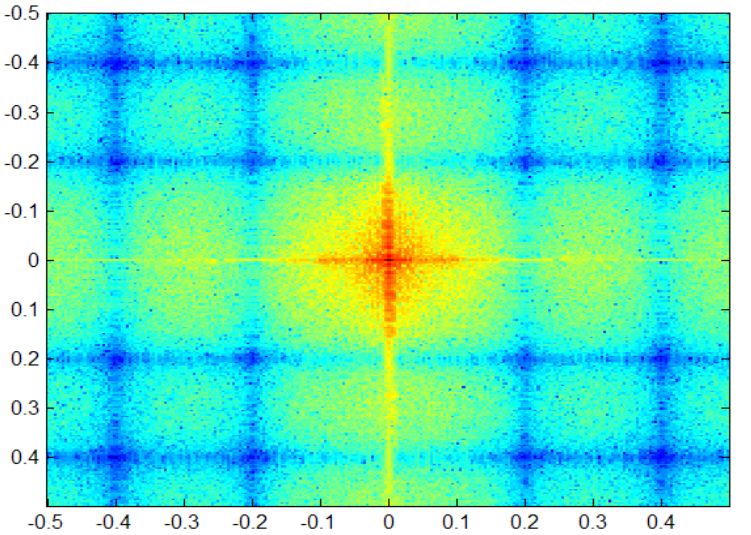
$\sigma = 2,5$

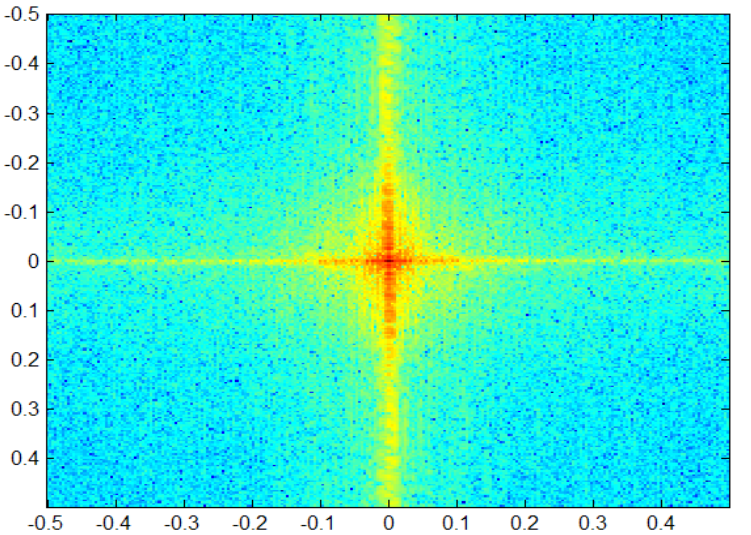
Exercice

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

```
clear, 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

Syntaxes et fonctions utiles

```
x=2;

u=[5 -3 2 4];
v=u+x;
w=v>1;

A=u'*u;
y=u*u';
B=A+x;
C=A+B;
D=A.*B;
E=D>2;
F=E(2:end,1:2);

t=A(:);
A=reshape(t,[4 4]);
```

size, whos
zeros, ones, double, uint8
imread, imwrite
figure, subplot, image, imagesc, imshow
colorbar, colormap, axis
load
cat
ginput, plot, line, hold, surf
min, max, abs, log10, sqrt, mod
fix
meshgrid
linspace, fft2, fftshift
conv2, freqz2
medfilt2
flipud
videoWriter, writeVideo, open