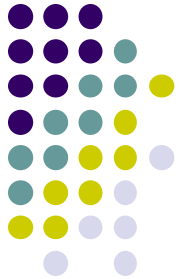
Introduction au Traitement des Images

Programmation dans l'environnement Matlab

Images à télécharger : https://sites.google.com/site/guillaumebourmaud/home/teaching

Organisation



- Cours
  - 5h20 (4×1h20)
- Enseignement intégré
  - 8h (3×2h40)
- Travaux pratiques
  - 6h40 (5×1h20), compte-rendu (Coeff 1)
- Projet
  - 13h20 (5×2h40), rapport + oral (Coeff 2)

Plan

- Image numérique
  - Définition
  - Lecture/Ecriture/Synthèse/Visualisation
- Filtrage
- Transformée de Fourier et Recouvrement fréquentiel (« Aliasing »)
- Espaces chromatiques

Image numérique (1/3)

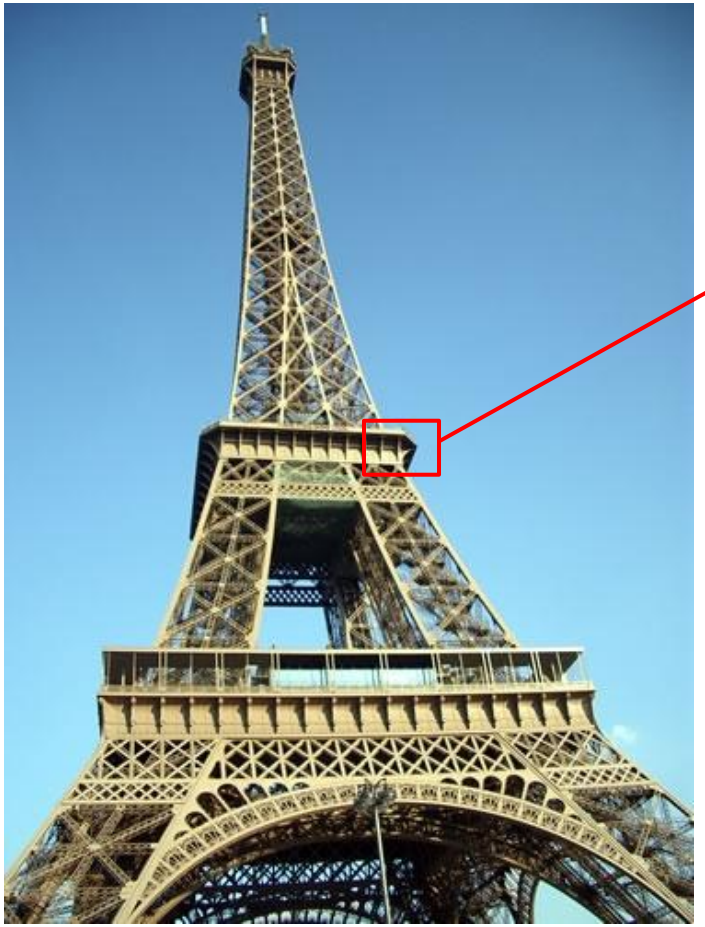


Image numérique
=
Matrice de pixels « colorés »

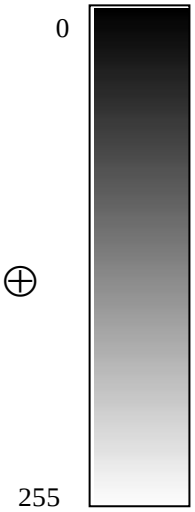
- Dimensions (nombre de pixels)
- Coordonnées (position du pixel)
- Valeur (couleur du pixel)

Image numérique (2/3)

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

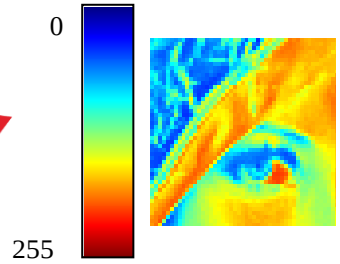
Intensité scalaire

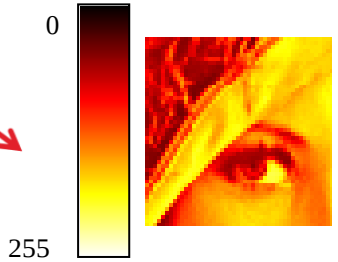
⊕

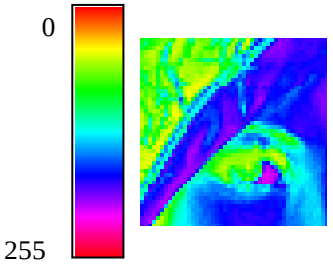


Palette
(niveaux de gris)

Autres palettes



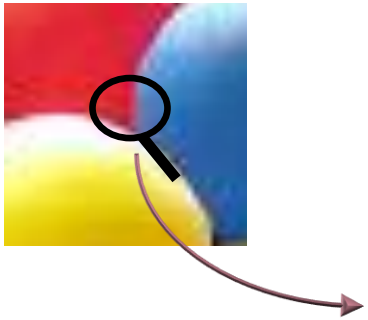




Couleurs indexées ou « fausses » couleurs

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

Image numérique (3/3)



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

Fonctions lecture/écriture/affichage

Lecture : imread, Ecriture : imwrite

Affichage : image, imagesc, imshow, surf

	image	imshow
Formats	uint8, uint16, double	uint8, uint16, double, …
Palette initiale	jet (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

Imagesc() = Image() + mise à l'échelle des intensités

Lecture et affichage
Fichier en « vraies couleurs »

```
>> A = imread('bdx.jpg');
>> whos
  Name          Size                    Bytes  Class    Attributes

  A           538x808x3                1304112  uint8

>> figure,imshow(A)
```

hauteur

largeur

« vraies » couleurs

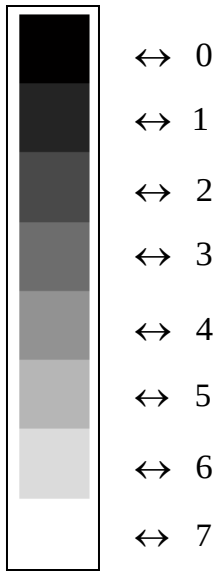
Lecture et affichage
Fichier en « fausses couleurs »

```
>> [A,palette] = imread('lena.bmp');
>> whos
  Name            Size                Bytes  Class     Attributes

  A             512x512            262144  uint8
  palette       256x3                6144  double

>> figure,imshow(A,colormap(jet))
>> figure,imshow(A,palette)
```

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, …

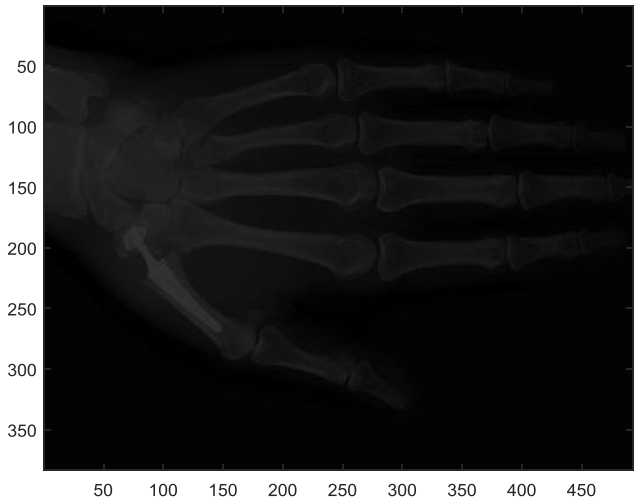
64 niveaux par défaut

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

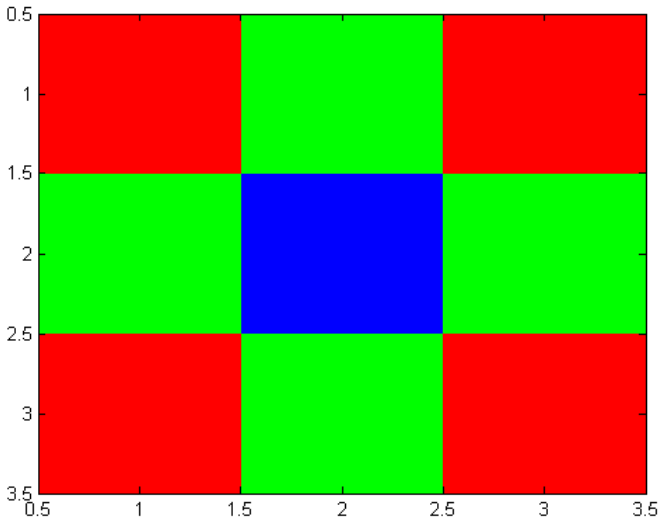
```
clear, close all

A=imread('radio.jpg');
figure, image(A)
colormap(gray(256))
colorbar
```




```
clear, close all

A=double(imread('radio.jpg'));
A1 = ((A-min(min(A)))/max(max(A)))*255;
figure, image(A1), colormap(gray(256)),
colorbar

figure, imagesc(A),
colormap(gray(256)), colorbar
```

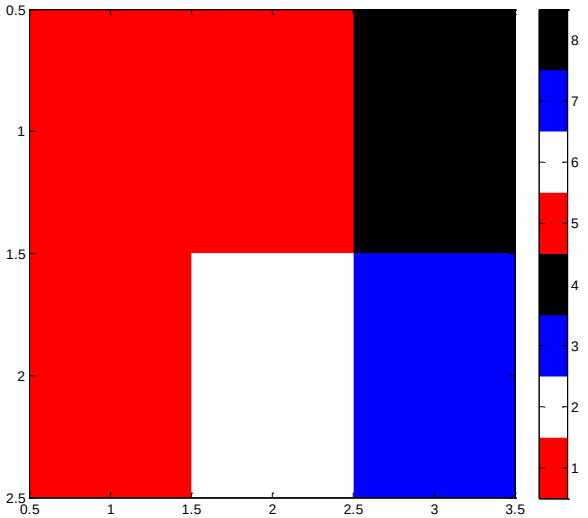
Synthèse en « vraie couleur »

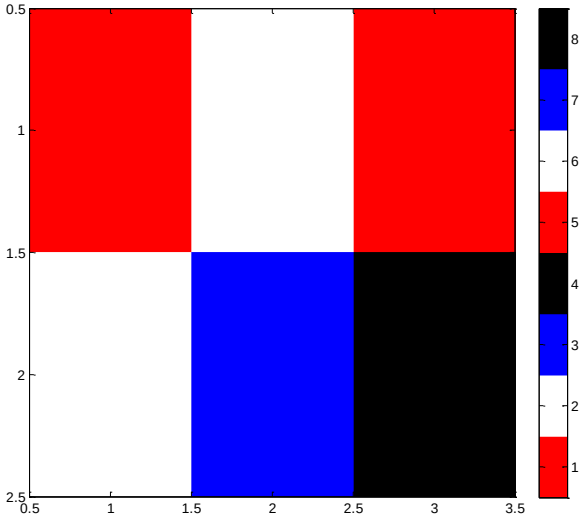
```
clear all
close all

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 couleur indexée

```
clear all
close all

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



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

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

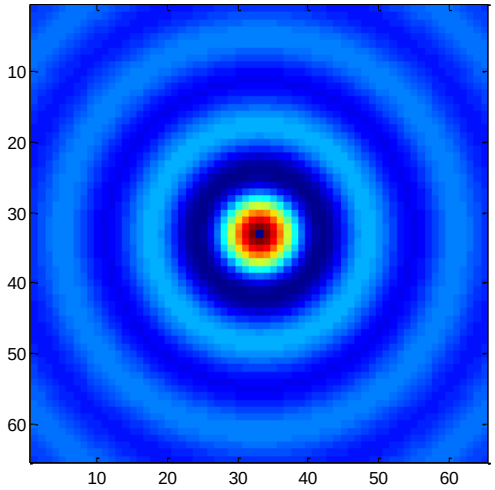


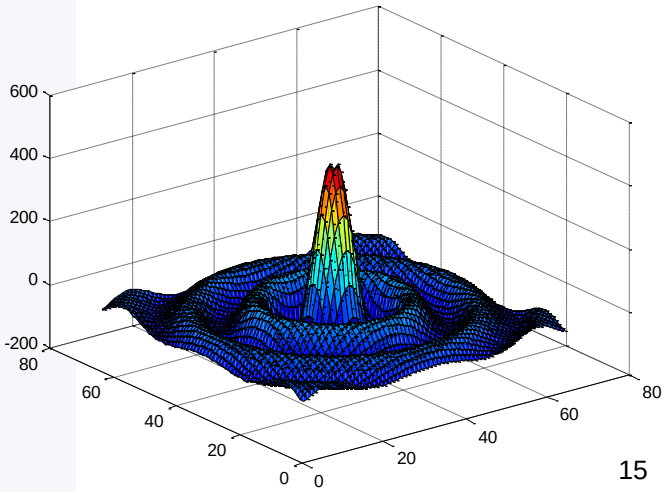


0 ou 1 !!!

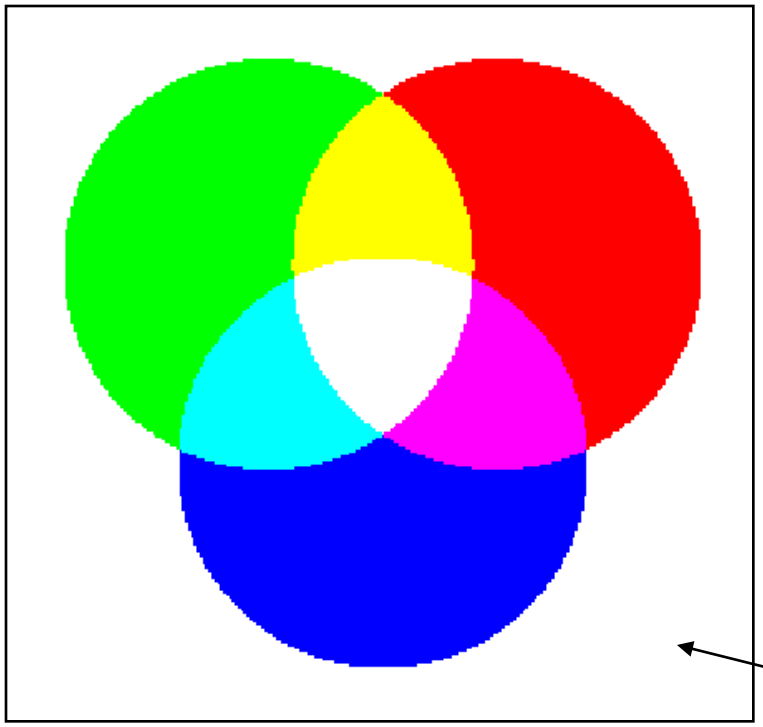
Synthèse analytique : meshgrid

```
clear, close all, clc
%solution longue (8 lignes)
x = -34:34; y = -32:32;
img1 = zeros(length(y),length(x));
for i=1:length(y)
   for j=1:length(x)
       r = sqrt(x(j)^2+y(i)^2);
       img1(i,j) = 1000*sin(r/2)/r;
   end
end
figure, imagesc(img1), axis square
figure, surf(img1)

%solution courte (3 lignes)
[X,Y]=meshgrid(-34:34,-32:32);
R=(X.^2+Y.^2).^0.5;
img2=1000*sin(R/2)./R;
figure, imagesc(img2), axis square
figure, surf(img2)
```





Exercice : synthèse additive



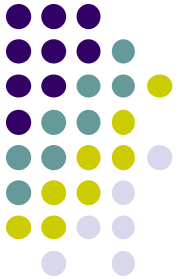
- « Vraie couleur »
- Couleur indexée

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

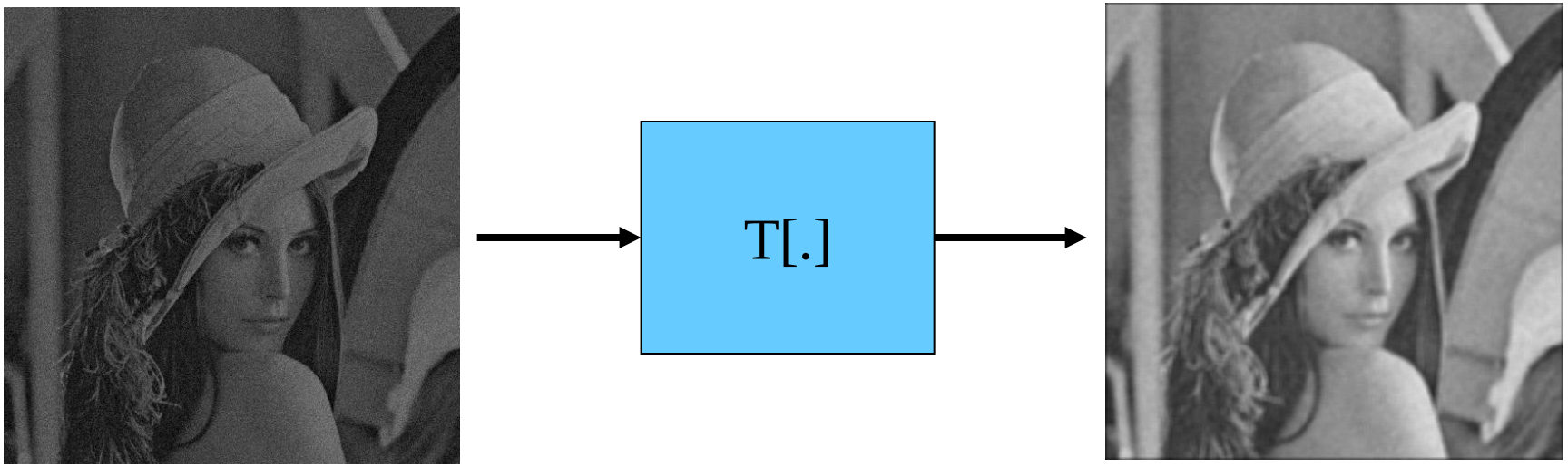


- Affichage de « lena.bmp » avec surf

```
figure, surf(A(1:4:end,1:4:end)), cameratoolbar;
```

- Affichage de « lena.bmp » avec imshow
- Capture d'un segment (**ginput**)
- Extraction du profil (signal horizontal ou vertical le plus « proche ») correspondant
- Affichage du profil

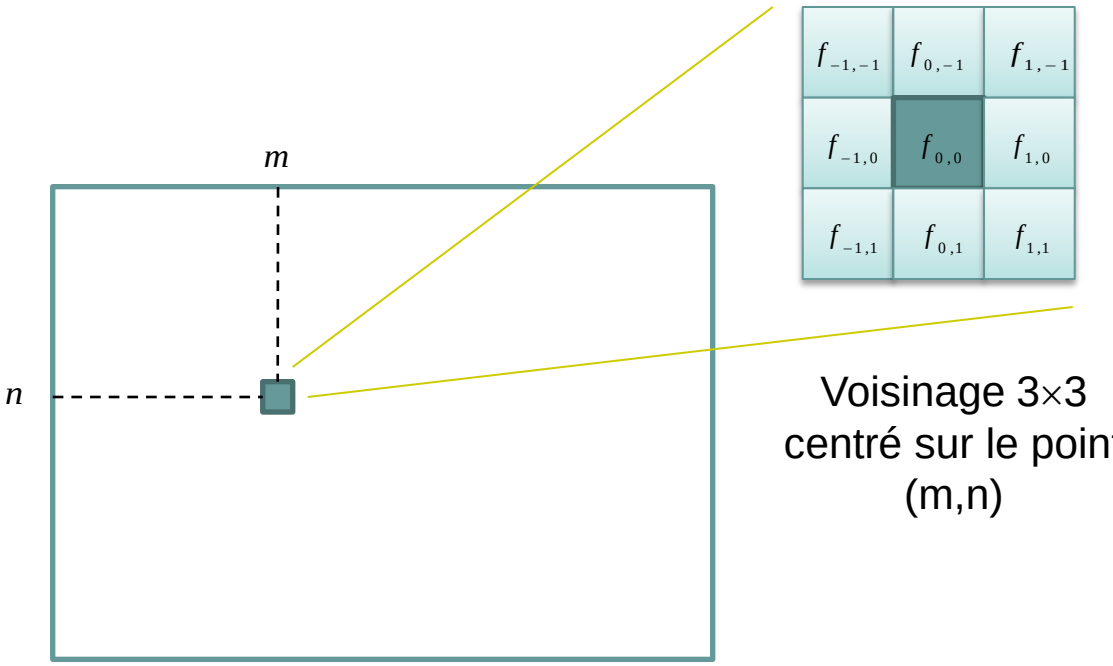
Filtres RIF



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

Convolution 2D

Implémentation



$$\begin{matrix} f_{-1,-1} & f_{0,-1} & f_{1,-1} \\ f_{-1,0} & f_{0,0} & f_{1,0} \\ f_{-1,1} & f_{0,1} & f_{1,1} \end{matrix}$$

$*$

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

$$\begin{matrix} h_{-1,-1} & h_{0,-1} & h_{1,-1} \\ h_{-1,0} & h_{0,0} & h_{1,0} \\ h_{-1,1} & h_{0,1} & h_{1,1} \end{matrix}$$

filtre 3×3

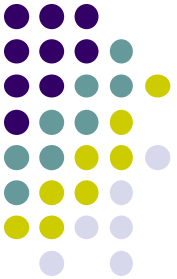
symétrie centrale

$$\begin{matrix} h_{1,1} & h_{0,1} & h_{-1,1} \\ h_{1,0} & h_{0,0} & h_{-1,0} \\ h_{1,-1} & h_{0,-1} & h_{-1,-1} \end{matrix}$$

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

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

Exemple : convolution 3×3



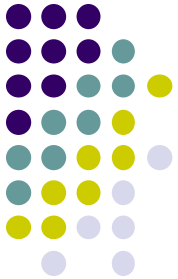
```
clear all
close all

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
```

Exemple : flouter un visage dans une image

```
clear, close all, clc;

A=double(imread('zidane.jpg'))/255;
figure, imshow(A);

[h,w,d]=size(A);
H=ones(40)/1600;
A_floue(:,:,1) = conv2(A(:,:,1),H,'same');
A_floue(:,:,2) = conv2(A(:,:,2),H,'same');
A_floue(:,:,3) = conv2(A(:,:,3),H,'same');

cx = 492;
cy = 190;
r = 140;
[X, Y] = meshgrid(1:w, 1:h);
mask = ones(h,w);
mask((X-cx).^2+(Y-cy).^2<r^2)=0;
figure, imshow(mask.*A + (1-mask).*A_floue);
```

Exemple : Détection de contours



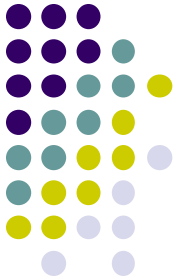
```
clear, close all, clc

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)))
```

$\sigma = 0{,}75$

$\sigma = 1{,}5$

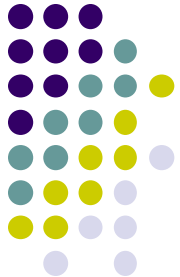
$\sigma = 2,5$

Exercice



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

Transformée de Fourier et recouvrement fréquentiel



Transformée directe

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

variables fréquentielles

variables spatiales

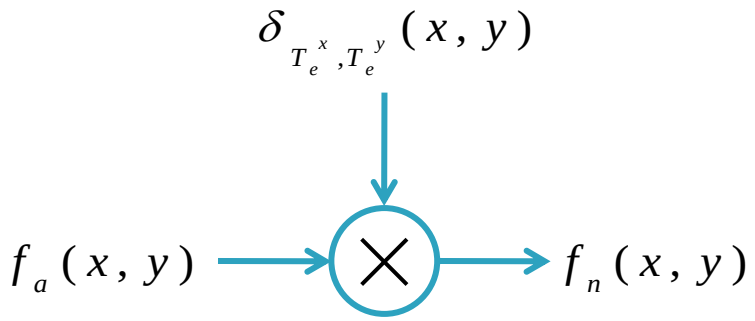
$$F(u,v) = \left|F(u,v)\right| e^{j\varphi(u,v)}$$

Transformée inverse

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

Echantillonnage 2D

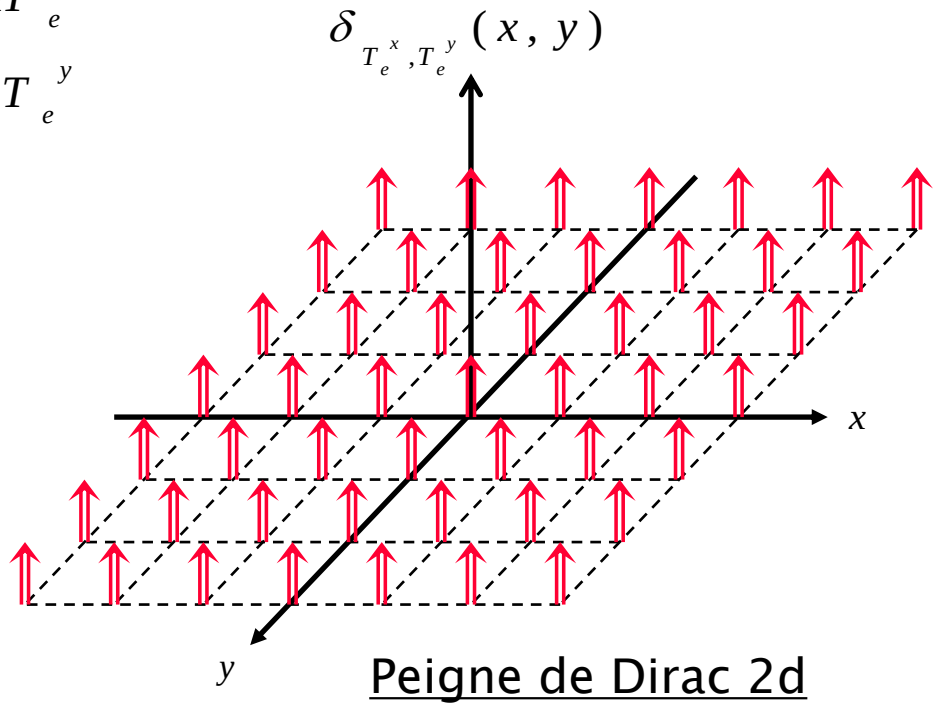
$$f_n(x,y) = \begin{cases} f_a(x,y) & \text{pour} \quad \begin{cases} x = mT_e^x \\ y = nT_e^y \end{cases} \\ 0 & \text{sinon} \end{cases}$$



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

séquence discrète

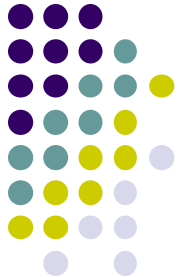
notée $f(mT_e^x, nT_e^y)$ ou $f(m,n)$



Peigne de Dirac 2d

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

Signaux discrets : TF(CD) et TFD



échantillonnage spatial

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

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

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

périodes d'échantillonnage

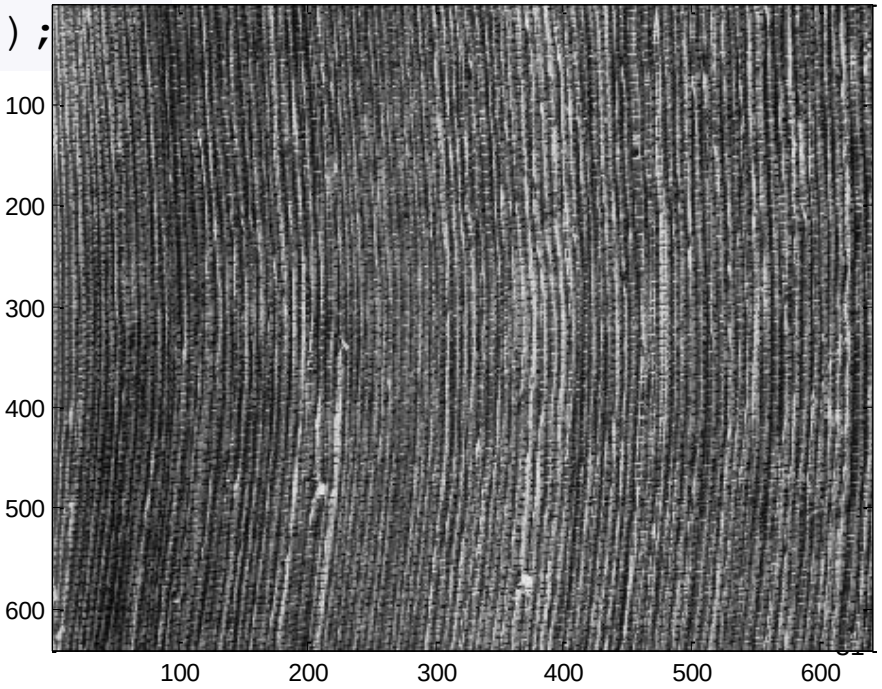
fenêtrage spatial et échantillonnage des fréquences

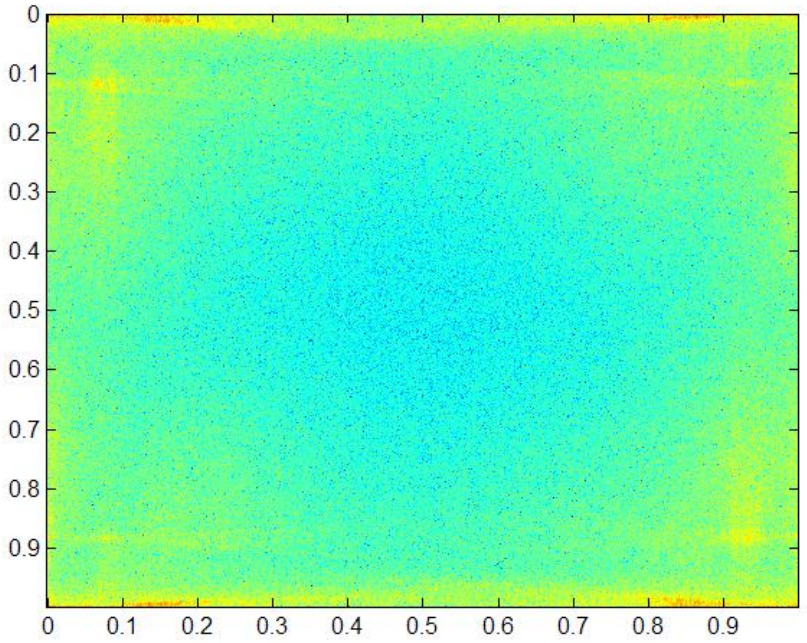
Transformée de Fourier discrète (d'une séquence discrète)
TFD

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

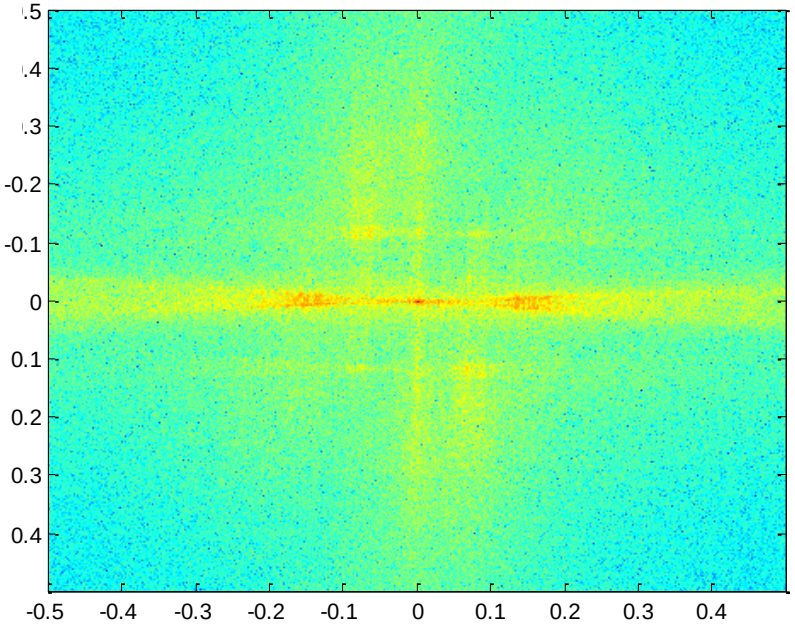
variables discrètes $\begin{cases} k \in [0, M-1] \\ l \in [0, N-1] \end{cases}$

```
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);
fx=linspace(-0.5,0.5-1/w,w); %w pair
fy=linspace(-0.5,0.5-1/h,h); %h pair
figure, imagesc(fx,fy,fftshift(B));
```





Spectre d'une image réelle



- Amplitude paire

$$\left|F\ (u\ ,v\ )\right| = \left|F\ (-u\ ,-v\ )\right|$$

- Phase impaire

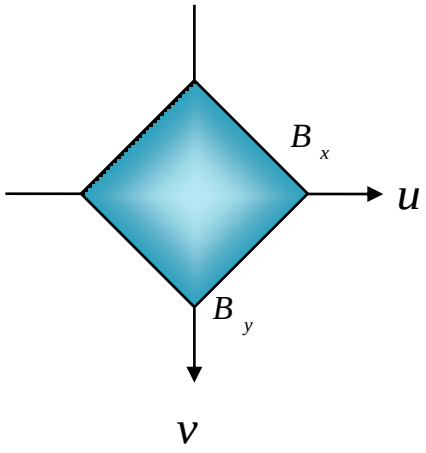
$$\varphi\ (u\ ,v\ ) = -\varphi\ (-u\ ,-v\ )$$

Exercice

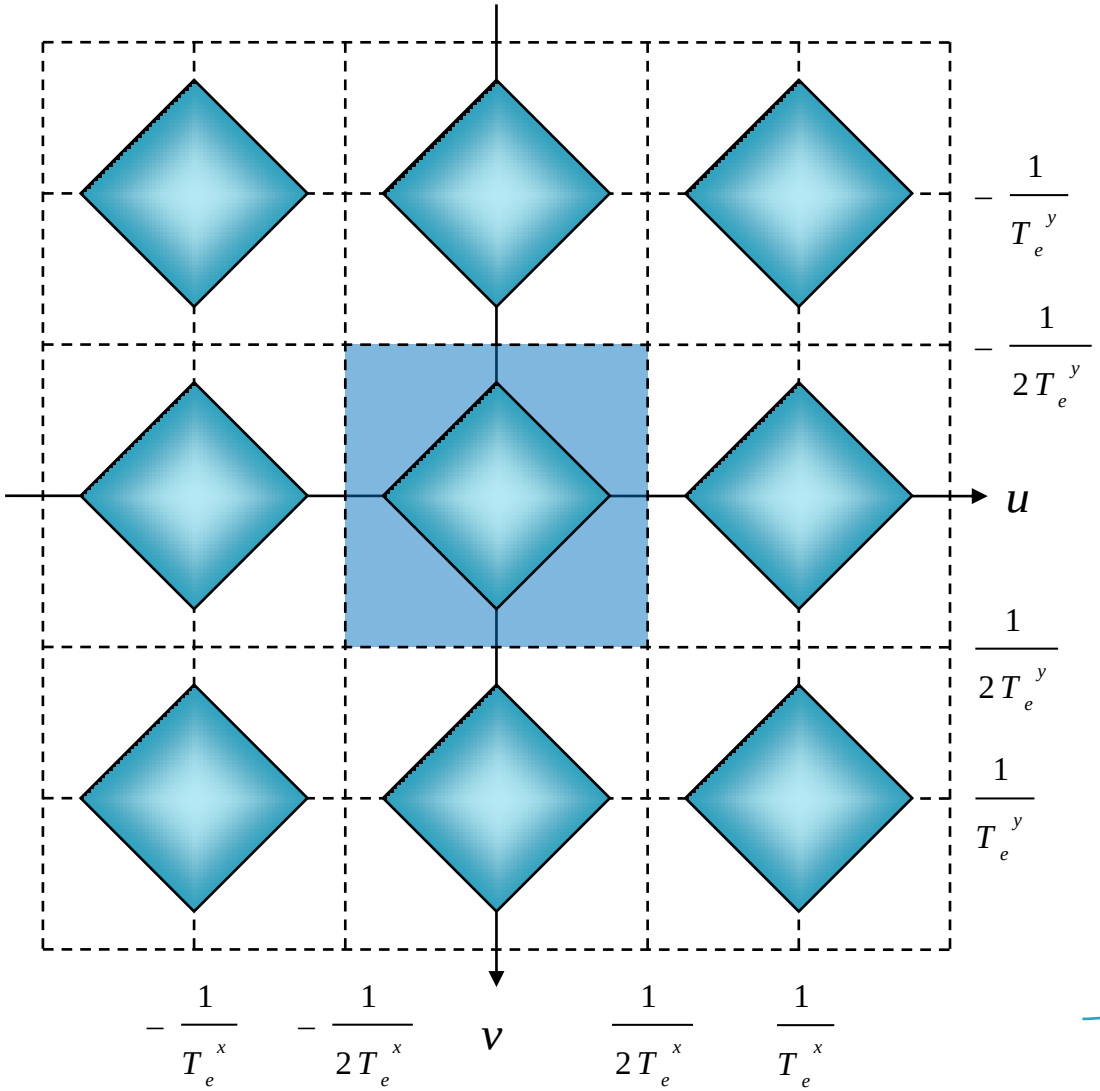
- Affichage de l'image 'monument_ext.bmp'
- Filtrage passe-bas pour éliminer/réduire le bruit
  - Justification du choix (visualisation de la TFD)
  - Vérification de l'adéquation des filtres (**freqz2**)
- Affichage des images filtrées et de leur spectre
- Calcul et affichage du « bruit » (différence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

Recouvrement fréquentiel (1/2)

$|F_a(u)|$



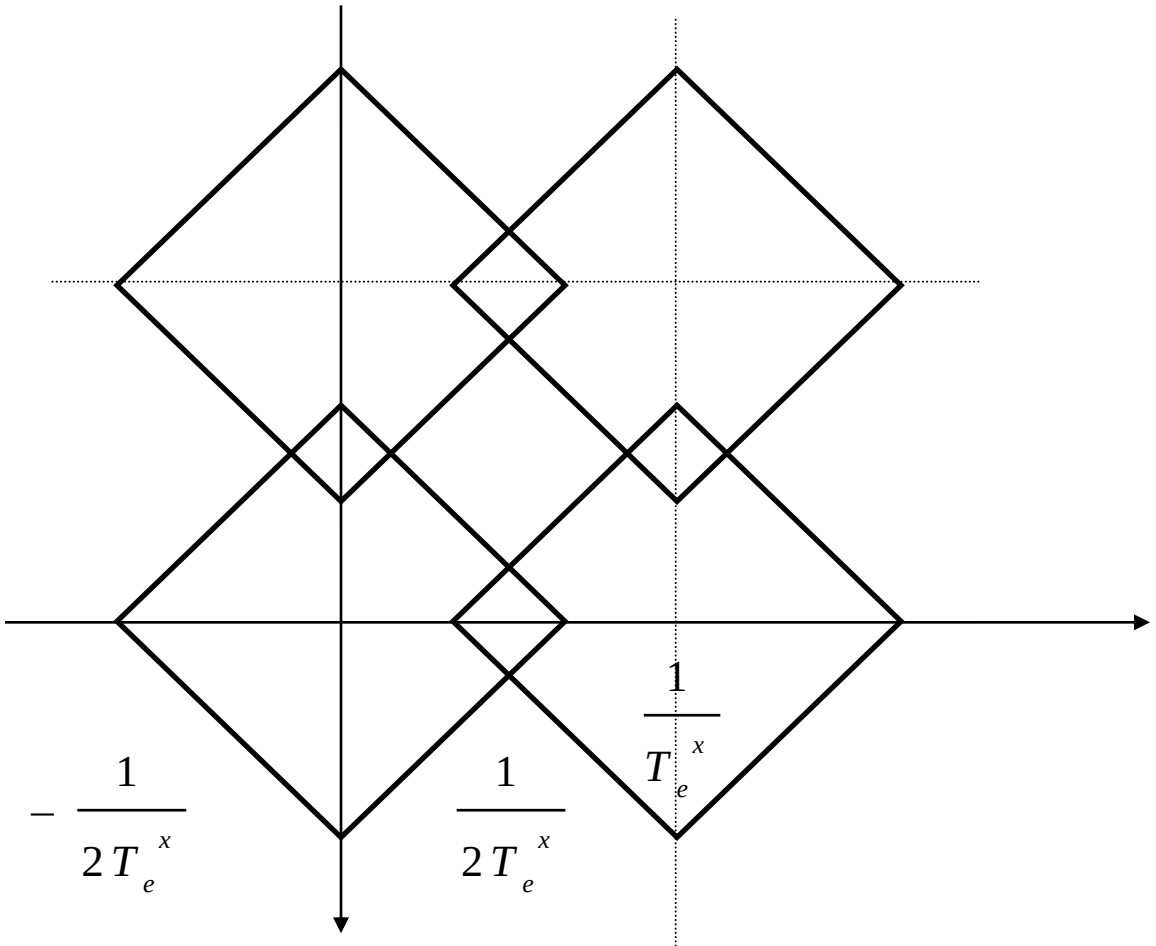
Signal 2d à bande passante limitée

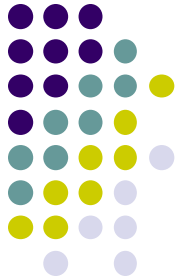


périodi-sation du spectre

$|F_n(u)|$

Recouvrement fréquentiel (2/2)





```
clear, close all, clc

A=imread('barbara.bmp');
figure
image(A)
colormap(gray(256))
axis('equal')
```

```
B=A(1:4:end,1:4:end);
figure
image(B)
colormap(gray(256))
axis('equal')
```

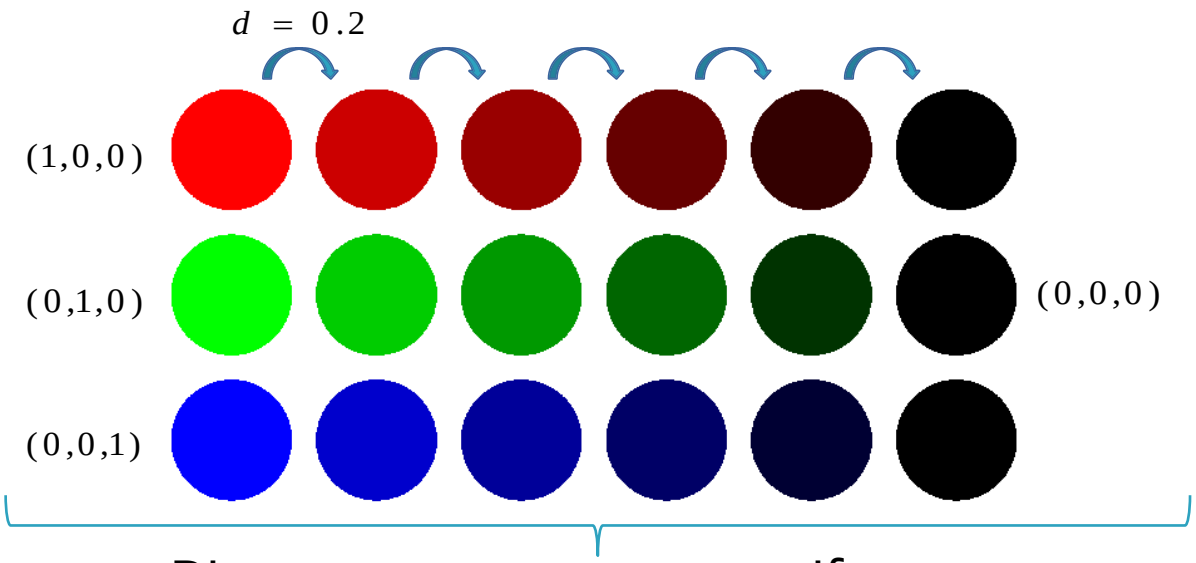
```
C=imresize(A,1/4);
figure
image(C)
colormap(gray(256))
axis('equal')
```


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

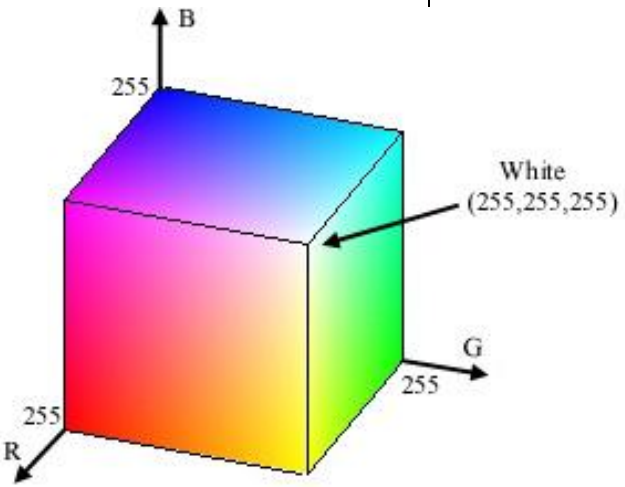

20
40
60
80
100
120
140
0
20
40
60
80
100
120
140
160
180


20
40
60
80
100
120
140
0
50
100
150

Espace chromatique RVB

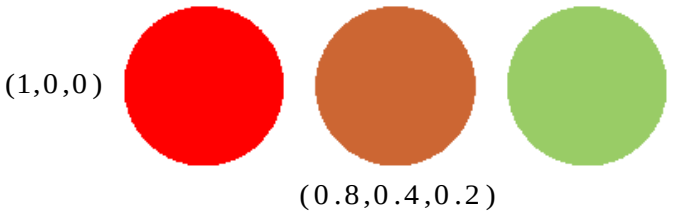


Distances perçues non « uniformes »



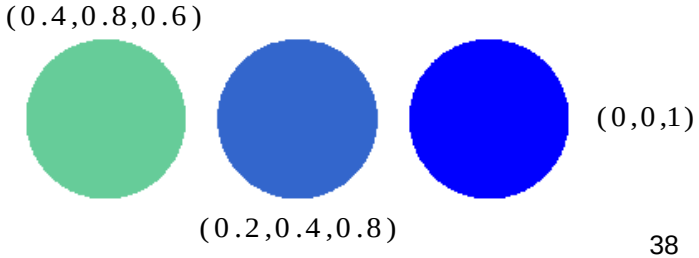
$$L = \frac{R + V + B}{3}$$

luminance (intensité moyenne)

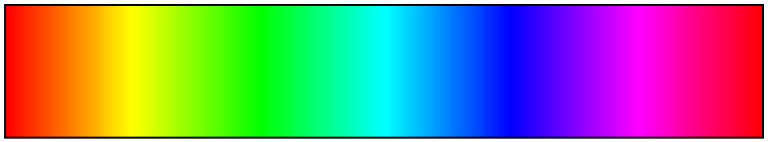


R, G, B ?

décomposition très difficile (même à 20% près)



Espace chromatique TSL (HSL)



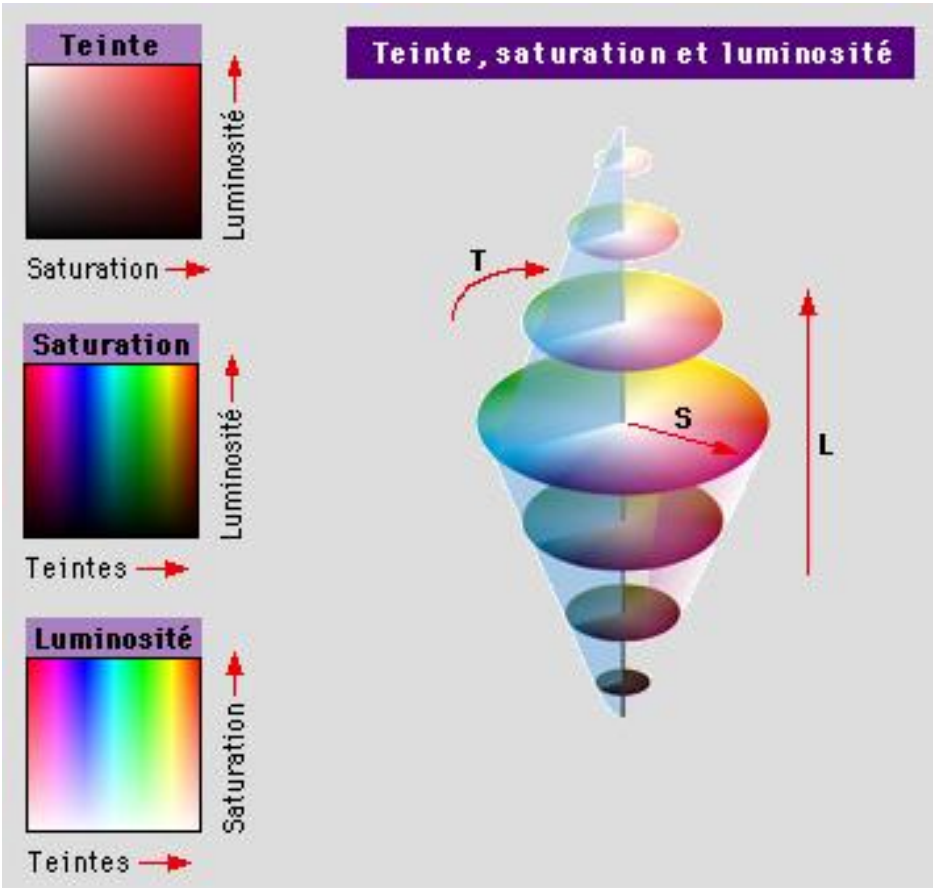
Teinte

Saturation

« mélange » avec le gris

Luminosité

« mélange » avec le noir



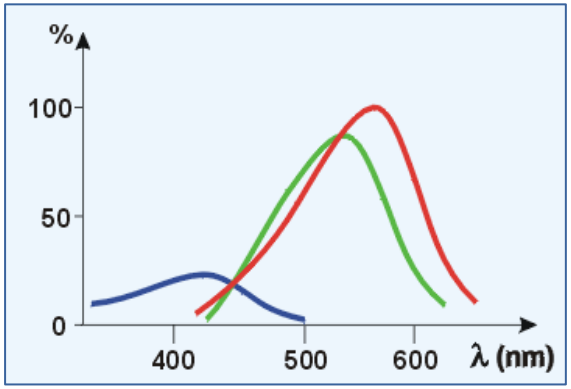
RVB : Luminance vs Chrominance

luminosité plus forte

contrastes plus faibles

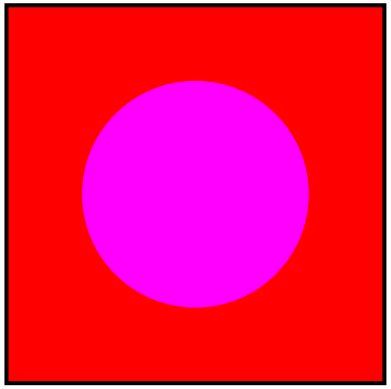
Niveaux de gris

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

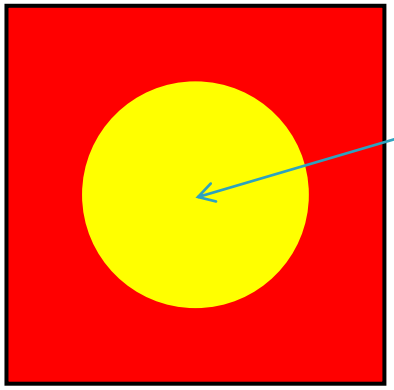


Sensibilité relative des photo-récepteurs de chrominance

Luminance ‘perceptuelle’

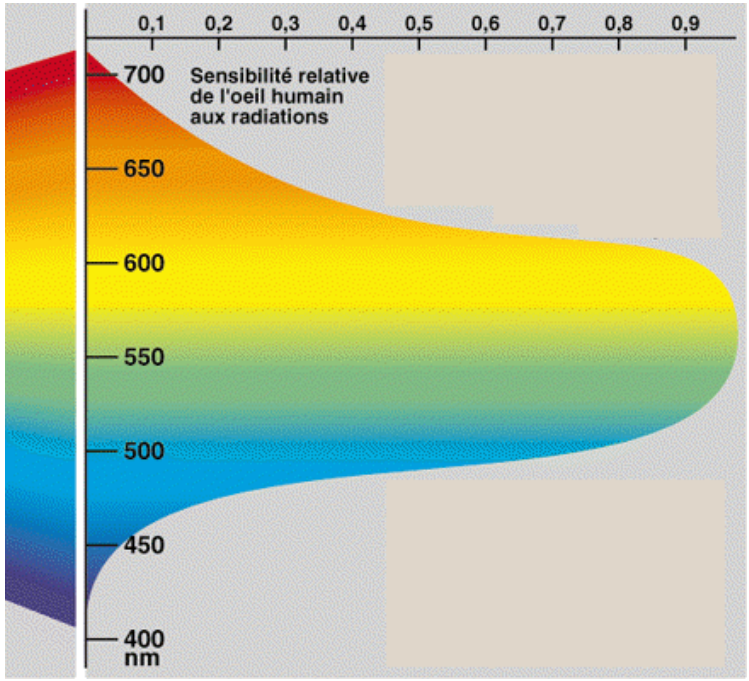


Rouge/Bleu



Rouge/Vert

luminosité
plus forte



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

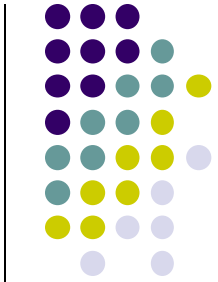
pondération plus faible

Espace colorimétrique YCbCr

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

$$C_b = 0.564\ (B - Y) + 128$$

$$C_r = 0.713\ (R - Y) + 128$$



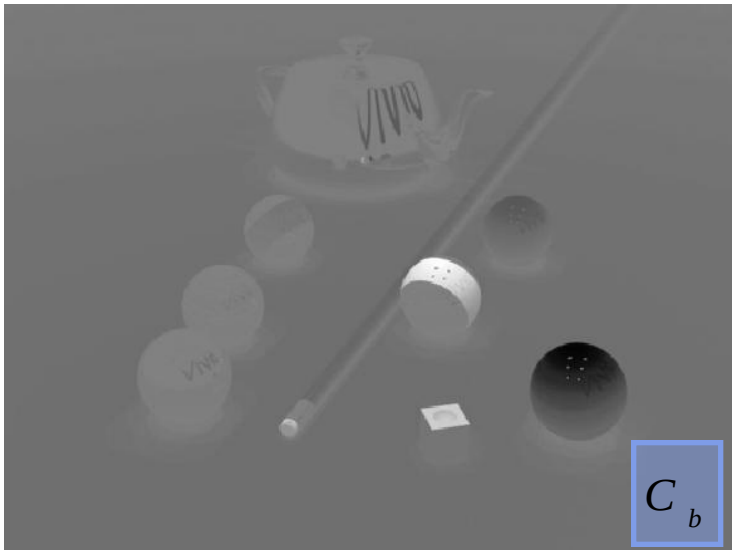
```
clear, close all, clc

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))
```

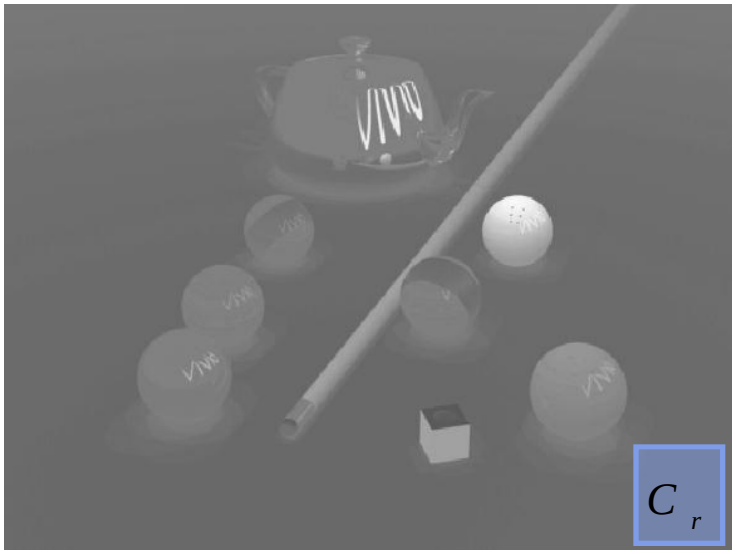

R

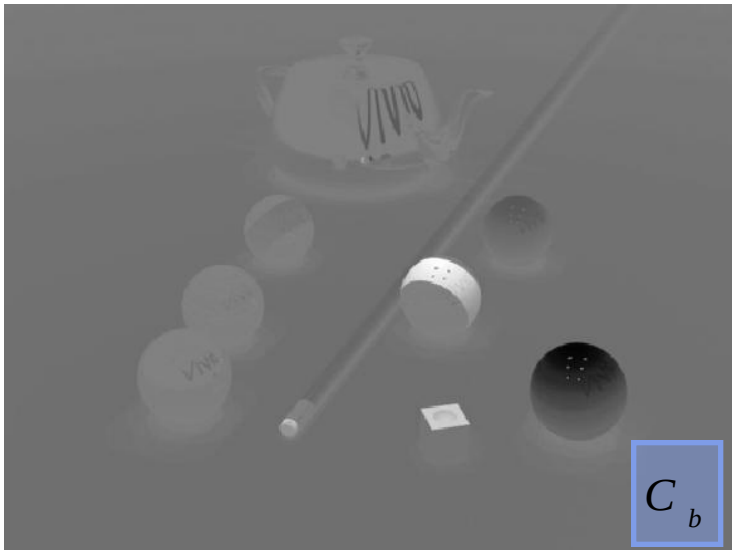

G


B

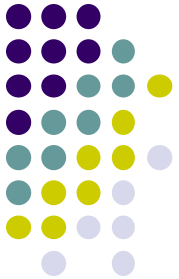

Y


C b


C r


C_b


B



FIN