



Traitement d'images numériques

T.D. 1 : éléments de TD

Récupérer sur edunao les images en format binaire Matlab (.mat) : 'foiemeta.mat', 'lena512.mat' puis 'barbara.mat'.

I. Acquisition et visualisation

Afficher les caractéristiques des images 'pout.tif', 'cameraman.tif', 'saturn.png' et 'kids.tif' (commande Matlab `imfinfo`).

Pour charger et visualiser les images monochromes on utilisera les commandes Matlab :

```
im=imread('nom.ext');
figure; imagesc(im); colormap(gray); colorbar
```

Pour charger et visualiser les images indexées on utilisera les commandes Matlab :

```
[im, palette]=imread('nom.ext');
figure; image(im); colormap(palette); colorbar
```

Pour charger et visualiser les images couleur RGB on utilisera les commandes Matlab :

```
im=imread('nom.ext');
figure; imagesc(im);
```

```
>> imfinfo('pout.tif')
```

```

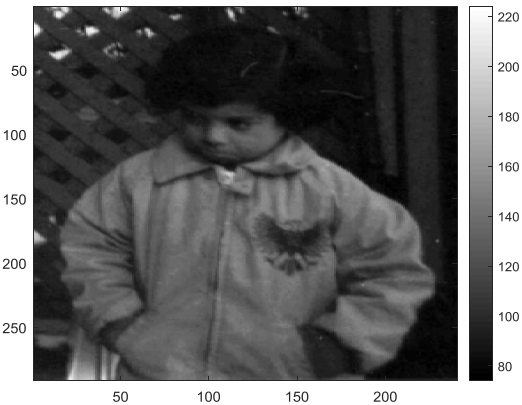
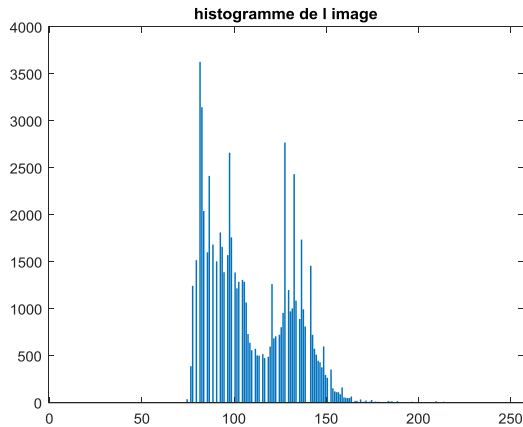
      Filename: 'C:\Program
Files\MATLAB\R2015a\toolbox\images\im...'
      FileModDate: '04-déc.-2000 12:57:50'
      FileSize: 69004
      Format: 'tif'
      FormatVersion: []
      Width: 240
      Height: 291
      BitDepth: 8
      ColorType: 'grayscale'
      FormatSignature: [73 73 42 0]
      ByteOrder: 'little-endian'
      NewSubFileType: 0
      BitsPerSample: 8
      Compression: 'PackBits'
      PhotometricInterpretation: 'BlackIsZero'
      StripOffsets: [8 7984 15936 23976 32089 40234 48335 56370
64301]
      SamplesPerPixel: 1
      RowsPerStrip: 34
      StripByteCounts: [7976 7952 8040 8113 8145 8101 8035 7931 4452]
      XResolution: 72
      YResolution: 72
      ResolutionUnit: 'Inch'
      Colormap: []
      GrayResponseUnit: 0.0100
      MaxSampleValue: 255
      MinSampleValue: 0
```

```
imfinfo('kids.tif')
    Filename: 'C:\Program
Files\MATLAB\R2015a\toolbox\images\imda...'
    FileModDate: '04-déc.-2000 12:57:58'
    FileSize: 95162
    Format: 'tif'
    FormatVersion: []
    Width: 318
    Height: 400
    BitDepth: 8
    ColorType: 'indexed'
    FormatSignature: [73 73 42 0]
    ByteOrder: 'little-endian'
    NewSubFileType: 0
    BitsPerSample: 8
    Compression: 'PackBits'
    PhotometricInterpretation: 'RGB Palette'
    StripOffsets: [1x16 double]
    SamplesPerPixel: 1
    RowsPerStrip: 25
    StripByteCounts: [1x16 double]
    XResolution: 72
    YResolution: 72
    ResolutionUnit: 'Inch'
    Colormap: [256x3 double]
    GrayResponseUnit: 0.0100
    MaxSampleValue: 255
    MinSampleValue: 0
```

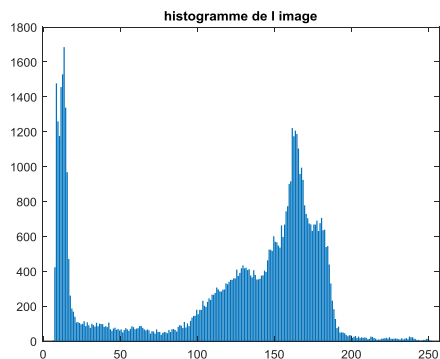
II. Analyse élémentaire

II.1. Déterminer et visualiser les histogrammes des images monochromes (on pourra utiliser la commande 1D de Matlab `hist`). Faites le lien entre les caractéristiques de l'histogramme, les indicateurs σ/μ , $(\max - \min)/(\max + \min)$ et le contraste de l'image. Pour les images couleurs, étudier les histogrammes marginaux dans l'espace RGB et HSV (on utilisera la commande `rgb2hsv` pour passer dans l'espace HSV).

```
%% Contraste des images monochromes
im=imread('pout.tif');
figure; imagesc(im); colormap(gray); colorbar
x=.5:1:255.5;
[Ne]=hist(double(im(:)),x); % histogramme
figure
bar(x,Ne)
% indicateurs
std(im1D)/mean(im1D)=0.21
(max(im1D) - min(im1D))/(max(im1D) + min(im1D)) =0.5
```



```
im=imread('cameraman.tif');
figure; imagesc(im); colormap(gray); colorbar
std(im1D)/mean(im1D)=0.53
(max(im1D) - min(im1D))/(max(im1D) + min(im1D)) =0.95
```

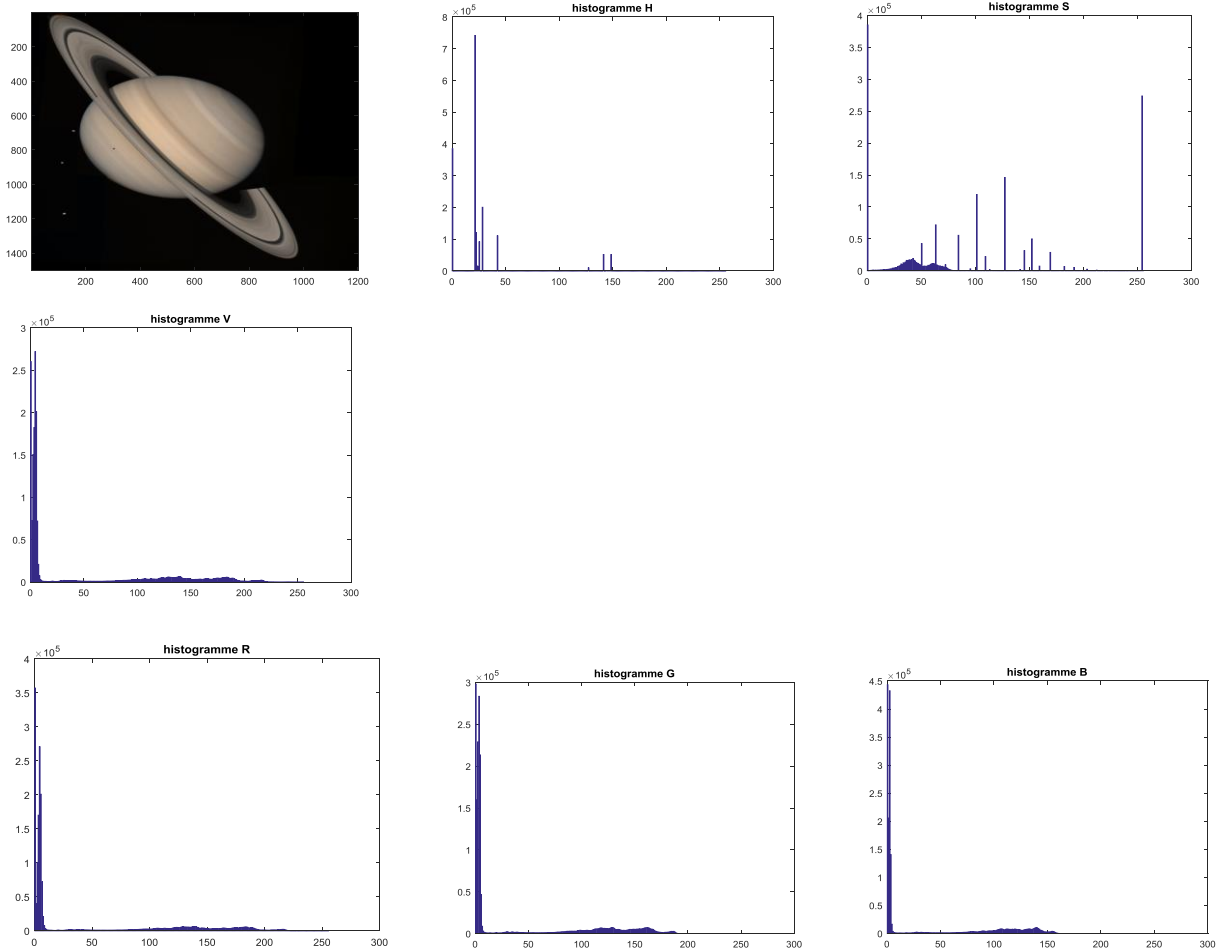


```

%% Contraste des images couleurs : colortype 'true color'
im=imread('saturn.png');
figure; imagesc(im);
imhsv=rgb2hsv(im);
% %% histogrammes marginaux pour chaque composante
% composantes RGB
imR=double(im(:,:,1));
imG=double(im(:,:,2));
imB=double(im(:,:,3));

% composantes HSV
imH=255*double(imhsv(:,:,1));
imS=255*double(imhsv(:,:,2));
imV=255*double(imhsv(:,:,3));

```



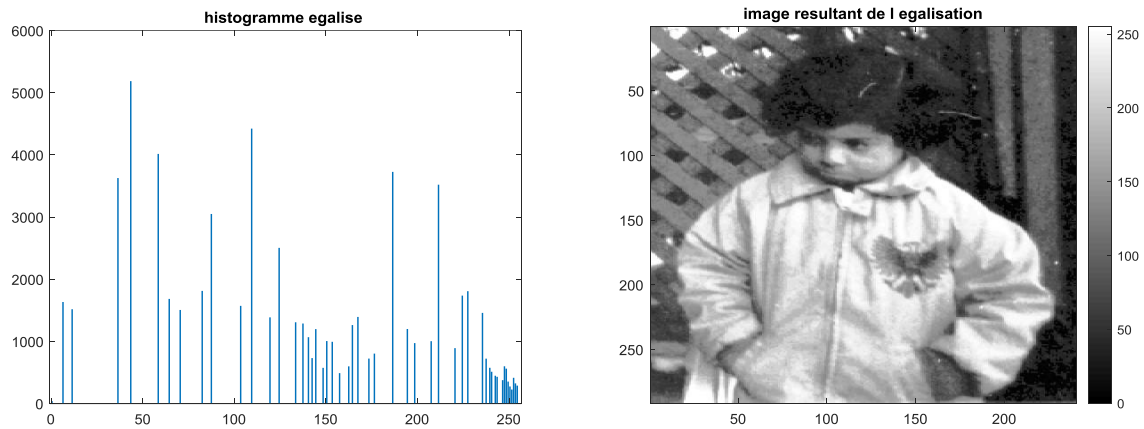
II.2. Proposer et mettre en oeuvre différentes transformations d'histogramme permettant d'améliorer le contraste des images monochromes 'pout.tif', 'cameraman.tif' et de l'image couleur 'saturn.png'.

L'égalisation d'histogramme permet d'obtenir théoriquement le meilleur contraste (par un histogramme plat) mais il amplifie le bruit présent dans l'image (un débruitage de l'image doit être utilisé au préalable).

La fonction à appliquer au pixel de l'image de valeur k est définie par : $(255/taille)*histcum(k)$, où $histcum(k)$ est l'élément de rang k de l'histogramme cumulé et $taille$ le nombre de pixels de l'image. Cette fonction est à implémenter (on peut utiliser pour cela la fonction `cumsum` de Matlab).

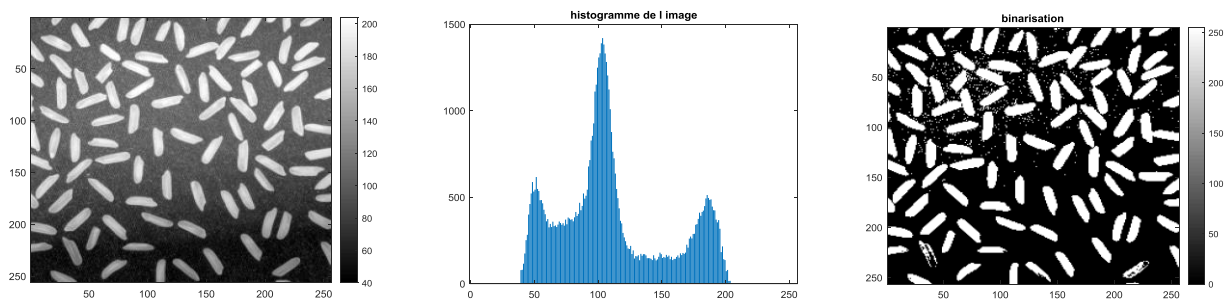
```
% %% égalisation d'histogramme
x=.5:1:255.5;
[N]=hist(double(im(:)),x); % histogramme
imegal=egal(double(im),N,x); % fonction « egal » pour l'égalisation
figure
imagesc(imegal,[0 255])
title('image resultant de l egalisation')
colormap(gray)
colorbar
x=.5:1:255.5;
[Ne]=hist(imegal(:),x);
figure
bar(x,Ne)
title('histogramme egalise ')
std(imegal(:))/mean(imegalD)=0.54
(max(imegal(:)) - min(imegal(:)))/(max(imegal(:)) + min(imegal(:)))=0.999

%% définition de la fonction d'égalisation d'histogramme: « egal »
function s=egal(e,N,x)
taille=size(e);
f=cumsum(N);
f1=255*f/(taille(1)*taille(2));
for i=1:taille(1),
for j=1:taille(2),
s(i,j)=f1(round(e(i,j)+1));
end;
end;
%% fonction matlab : histeq
imegal=histeq(double(im)/max(im(:)));
```



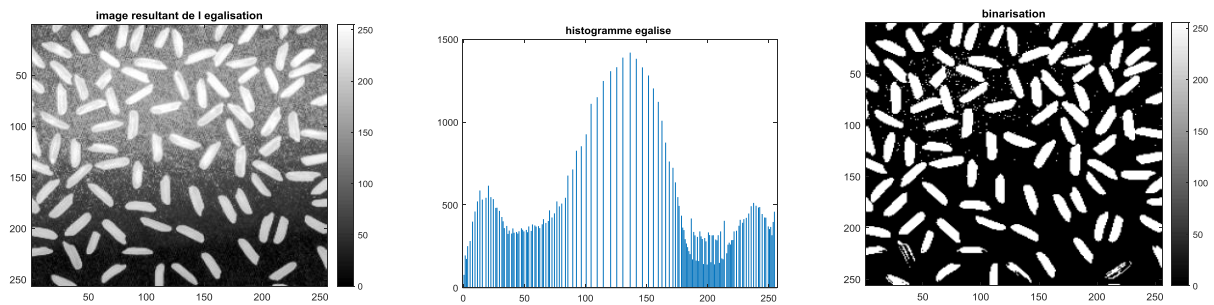
II.3. Réaliser une binarisation de l'image 'coins.png' et 'rice.png' par seuillage. Etudier l'influence du seuil de binarisation sur l'image obtenue.

```
%% binarisation rice : distortion fond -> choix seuil impossible
imb=double(im)>120; %78.6
figure
imagesc(imb*255,[0 255])
title('binarisation')
colormap(gray)
colorbar
```



```
imb=double(imegal)>185; %172
figure
imagesc(imb*255,[0 255])
title('binarisation')
```

```
colormap(gray)
colorbar
```



III. Numérisation

III.1. Quantifier sur b bits l'image monochrome du fichier 'foiemeta.mat', Sélectionner la région d'intérêt (ROI) correspondant au foie et visualiser l'image d'origine et l'image quantifiée. On pourra utiliser la fonction Matlab `round`. Conclure sur l'effet visuel de la quantification pour b variant de 8 à 5. Vérifier les caractéristiques de l'erreur de quantification (histogramme, moyenne, puissance) sur la ROI.

L'erreur de quantification suit une loi uniforme entre $-q/2$ et $+q/2$ de moyenne

$$q = \frac{v_{\max} - v_{\min}}{2^b - 1} = \frac{255}{2^b - 1}$$

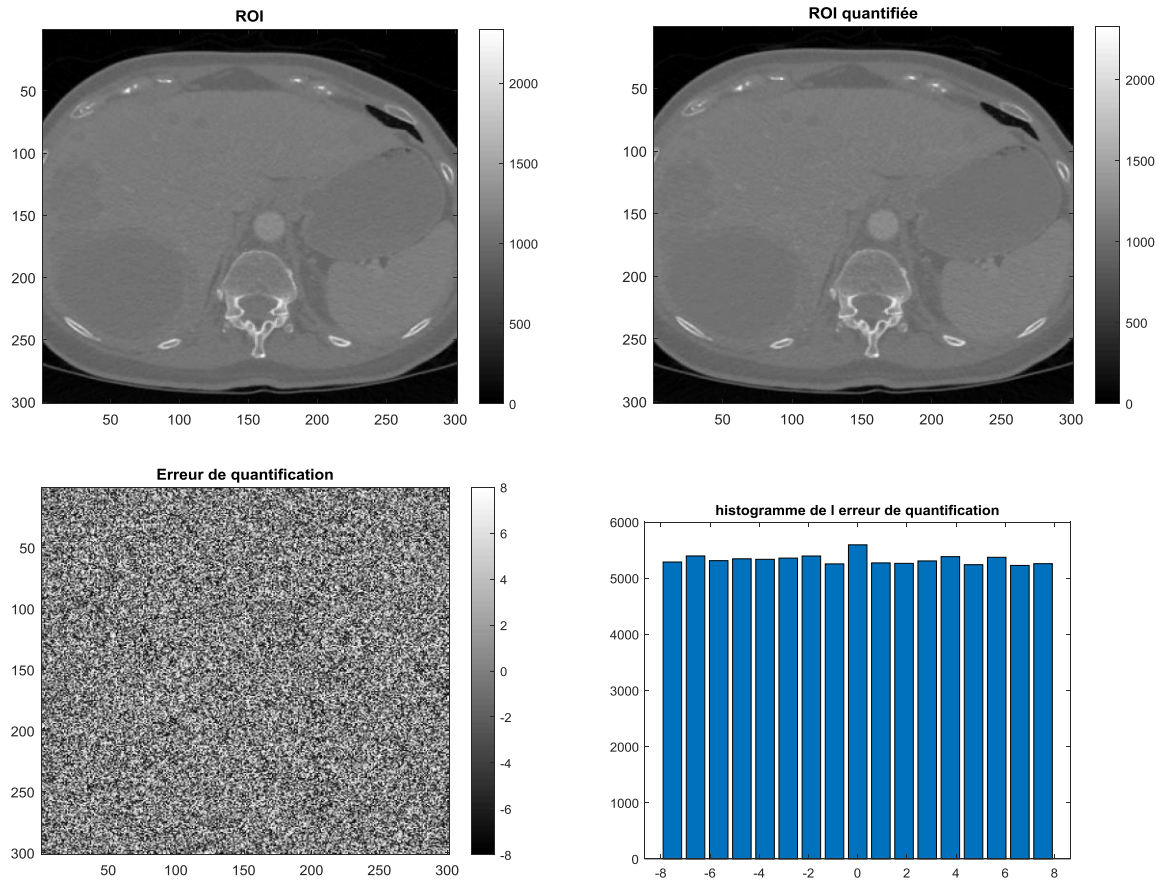
nulle et de puissance $(q^2)/12$ où

```
% quantification
load foiemeta
imagesc(foiemeta); colormap(gray);colorbar
b=8; % nombre de bits de quantification
q=(max(max(foiemeta))-min(min(foiemeta)))/(2^b-1) % pas de quantification
q=17
10*log10((q^2)/12)=13.8 dB % puissance du bruit de quantification
```

```
foieq=q*round(foiemeta/q); % quantification de l'image
% ROI
figure;imagesc(foiemeta(100:400,100:400)); colormap(gray);colorbar
figure;imagesc(foieq(100:400,100:400)); colormap(gray);colorbar
bruitq=foieq-foiemeta;
figure;imagesc(bruitq(100:400,100:400)); colormap(gray);colorbar
```

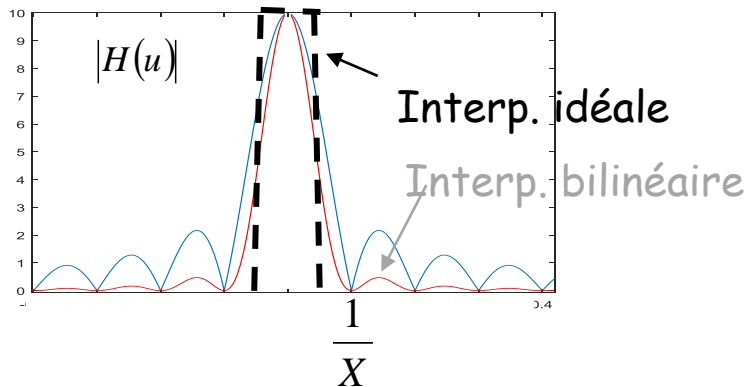
```
% histogramme de la ROI
bruitq=bruitq(100:400,100:400);
imlD=double(bruitq(:));
[N,x]=hist(imlD,round(q));
figure
bar(x,N) ;title('histogramme de l'erreur de quantification')
```

$\text{Moyenne_estimée} = \text{mean}(\text{im1D}) = -0.0198 = \text{Moyenne théorique (nulle)}$
 $\text{Puissance_estimée} = 20 \cdot \log_{10}(\text{std}(\text{im1D})) = 13.78 \text{ dB} = \text{Puissance théorique}$
 $(q^2/12) \text{ (dB)}$



III.2. Sous-échantillonner l'image contenue dans le fichier 'lena512.mat' puis 'barbara.mat' (par prélèvement d'un pixel sur quatre : $y_4 = y(1:4:\text{end}, 1:4:\text{end})$). Proposer et mettre en œuvre différentes techniques d'interpolation pour reconstituer l'image d'origine (on utilisera la commande `imresize`). Conclure.

L'interpolation au plus proche voisin, consiste à augmenter la taille de l'image en dupliquant les pixels voisins de chaque pixel. Elle conduit à une image «pixélisée» à hautes fréquences (HF). Cet effet est visible au voisinage des variations d'intensité de l'image (contours). L'interpolation bilinéaire atténue plus fortement les HF en dehors de la bande de fréquence de l'image (celle de l'interpolation idéale).



```
%% interpolation
load lena512.mat
X=lena512;
scale=4 % facteur de sous-échantillonnage
figure;imagesc(X); title('Lena');colormap(gray);colorbar
X2=X(1:scale:end,1:scale:end);
figure;imagesc(X2);title('Lena sous-échantillonnée'); colormap(gray);colorbar
X22n=imresize(X2,scale,'nearest');
figure;imagesc(X22n);title('Lena sous-échantillonnée interpolée : plus proche
voisin'); colormap(gray);colorbar
X22bl=imresize(X2,scale,'bilinear');
figure;imagesc(X22bl);title('Lena sous-échantillonnée interpolée: bilinéaire');
colormap(gray);colorbar
X22bc=imresize(X2,scale,'bicubic');
figure;imagesc(X22bc);title('Lena sous-échantillonnée interpolée: bicubique');
colormap(gray);colorbar
```

