Travaux pratiques et travaux dirigés de traitement d'images numériques Séances 1, 2, 3, 4 et 5 Institut Galilée 2013-2014

G. Dauphin et A. Beghdadi

Séance 1

(47)

Ce polycopié contient à la fois les exercices à réaliser pendant les séances de travaux dirigés et les énoncés à réaliser pendant les travaux dirigés. Le fait de réunir les deux dans un même polycopié permet de mieux mettre en évidence le fait que les travaux dirigés portent sur des notions qui sont vues en travaux pratiques. Les éléments étudiés portent sur les domaines suivants du traitement d'image.

- Acquisition et représentation du signal image : EX 1, 2, 3, 4, 5, 6, 24, 25, 30 TP 1, 4, 5, 13, 4 (séance 1 et 2).
- Analyse statistique du signal image : TP 5, EX 6, 14 TP 6 (séance 2 et 3).
- Analyse fréquentielle du signal image: EX 7, 9, 10, 11, 12, 20 TP 6, 8, 11, 13, 14 (séance 2, 3 et 4)
- Filtrage: EX 12, 15, 16, 19, 20, 21, 26, 28 TP 24, 25, 26, (séance 3, 4 et 5).
- Amélioration et restauration d'une image bruitée :EX 26, 27 TP 17, 19 (séance 3 et 5).
- Segmentation: EX 22, TP 27, 28, 30, 31, 32, 33 (séance 5).
- Couleurs: EX 1, 17, 18 TP 21, TP 22, 23,(séance 1 et 4)
- Compression: EX 29 (hors programme).

Une version électronique est disponible dans

http://www-12ti.univ-paris13.fr/~dauphin/IM/tp_tin_ir.pdf

Travaux pratiques

Explications sur les compte-rendu à rendre

(44)

A la fin de chaque séance, il y a un compte-rendu à rendre. Ce compte-rendu est à envoyer par mail sous format pdf. Pour chaque questions il doit contenir les éléments suivants :

- Numéro de la question
- Ligne de code Matlab permettant de générer l'image
- Observations sur l'image et commentaires sur la question
- Image réalisée par simulation, cette image doit s'afficher avec une taille suffisamment importante pour pouvoir bien visualiser ce qui est important.

Le compte-rendu doit porter sur l'ensemble de la séance sauf pour la première séance, où il ne porte que sur les TP 2, 3, p. 7, 8.

Quelques indications sur Matlab

(1)

Cours 1 Les images binaires sont formées de 0 et de 1. Ils permettent de stocker des images formées des teintes noires (0) et blanches (1).

```
im1=imshow('text.png');
figure(1); imshow(im1);
```

Dans cet exemple im1 indique qu'en mémoire est stockée une matrice dont les composantes sont 0 ou 1. Le fait qu'il n'y ait que deux valeurs est la raison pour laquelle on appelle cela une image binaire. Les commandes suivantes permettent d'abord de mettre sous la forme d'un vecteur vertical les composantes de l'image binaire puis de supprimer les doubles dans ce vecteur (i.e. les valeurs identiques n'apparaissent qu'une fois).

```
val=im1(:);
unique(val),
```

Les images en niveaux de gris permettent d'afficher des images qui contiennent différentes teintes entre le noir et le blanc.

```
im2=imread('trees.tif');
figure(1); imshow(im2);
```

Elles sont mémorisées sous la forme d'une matrice dont les composantes peuvent des entiers entre 0 et 255. Les commandes suivantes permettent d'afficher une courbe des les valeurs prises sont les valeurs des pixels de l'image lorsqu'on parcourt cette image de haut en bas puis de gauche vers la droite.

```
val=im2(:);
figure(1); plot(val);
```

Les images en niveaux de gris peuvent aussi être composées de réels entre 0 et 1.

```
im2Bis=double(im2)/255;
figure(1); imshow(im2Bis);
val=im2Bis(:);
figure(2); plot(val);
```

Les images colorées sont parfois mémorisées sous la forme d'un tableau de chiffres chaque chiffre correspondant à un index dans une table de couleurs. Les commandes suivantes permettent par exemple de récupérer l'image trees.tif.

```
[im3,map]=imread('trees.tif');
im3Bis=ind2rgb(im3,map);
figure(1); imshow(im3Bis);
```

im3 désigne une matrice dont les composantes ne sont pas des teintes. En effet si on cherche à les afficher on obtient une image qui n'a pas de sens (pourquoi les parties rouges auraient-elles la même teinte que les parties noires ?).

```
figure(2); imshow(im3);
```

map désigne un tableau composé de trois colonnes, la première correspond aux composantes de rouge, la deuxième aux composantes de vert et la troisième aux composantes de bleu. La commande suivante donne la taille de map, la première valeur correspond au nombre de lignes et la deuxième valeur correspond au nombre de colonnes.

```
size (map),
```

Si on annule les composantes de la première colonne, l'image reconstruite n'aura plus de rouge.

```
mapBis=map;
mapBis(:,1)=0;
im3Ter=ind2rgb(im3,mapBis);
figure(3); imshow(im3Ter);
```

Ces images couleurs sont stockées sous la forme d'une matrice (ici im3) dont les composantes indiquent pour chaque position des pixels un index vers une table (ici map) où sont référencées les couleurs associées à cet index (voir exercice 1). Les commandes suivantes illustrent ce fonctionnement en transférant l'index 127 de im3 à l'index 0.

```
im3Qua=im3; im3Qua(im3==127)=0;
im3Qui=ind2rgb(im3Qua,map);
figure(4); imshow(im3Qui);
```

Pour comprendre ce que l'on observe, on constate d'abord que la première ligne de map référence la teinte noire tandis que la ligne 128 référence une teinte blanche.

```
map(1,:),
map(128,:),
```

Cela permet de comprendre pourquoi les parties de l'image couleur qui étaient très blanches (associé à l'index 127) sont alors devenues noires (associées à l'index 255).

Les images peuvent aussi être mémorisées sous la forme d'une matrice-3D formées chacune d'entiers de 0 à 255 (par exemple image (1,1,1) désigne l'intensité du rouge [3ème 1] présent dans le pixel qui est sur la colonne 1 [1er 1] et sur la ligne 1 [2ème 1]). Les commandes suivantes permettent par exemple de charger et visualiser l'image autumn.tif, elles permettent aussi de lire sur une courbe les valeurs des composantes couleurs en parcourant l'image de haut en bas, puis de gauche vers la droite puis du rouge vers le vert et enfin le bleu.

```
im4=imread('autumn.tif');
figure(1); imshow(im4);
val=im4(:);
figure(2); plot(val),
```

La taille d'une image est le nombre de lignes et le nombre de colonnes. La commande suivante indique dans son premier chiffre le nombre de ligne, dans son deuxième chiffre le nombre de colonnes et dans son troisième chiffre 3 s'il y a effectivement trois composantes couleurs.

```
size(im4),
```

Le traitement d'image amène à faire des opérations sur les valeurs de chaque pixel. Il faut donc que ces valeurs soit un type adapté appelé double en Matlab. La convention est alors que les valeurs doivent être comprises entre 0 et 1.

```
im4Bis=double(im4)/255;
figure(3); imshow(im4Bis);
val=im4Bis(:);
figure(4); plot(val),
```

Pour vérifier si une image est au format entier entre 0 et 255 (noté ici uint8) ou au format double, on peut utiliser la commande class

```
class(im4),
class(im4Bis),
```

On peut aussi vérifier les valeurs prises par les images avec

```
max(max(max(im4))),
min(min(min(im4))),
max(max(max(im4Bis))),
min(min(min(im4Bis))),
```

On peut former une image plus petite en ne considérante qu'un ligne sur deux et une colonne sur deux (c'est-à-dire un pixel sur quatre).

```
im7=im4(1:2:end,1:2:end,:);
figure(1); imshow(im4);
figure(2); imshow(im7);
```

Une façon plus générale de modifier la taille d'une image peut se faire avec les commandes suivantes :

```
im8=imresize(im2,[256 256]);
figure(1); imshow(im2);
figure(2); imshow(im8);
```

Cette commande ne fonctionne pas en tant que telle pour les images couleurs. Pour réaliser cette transformation sur une image couleur, il suffit que cette image couleur soit d'abord convertie de façon à avoir des valeurs entre 0 et 1, puis de définir avec zeros une image couleur de la taille souhaitée, et enfin pour chaque composante de la remplir avec le résultat de l'application de la commande imresize sur chaque composante couleur.

```
im10=zeros(256,256,3);
im10(:,:,1)=imresize(im4Bis(:,:,1),[256 256]);
im10(:,:,2)=imresize(im4Bis(:,:,2),[256 256]);
im10(:,:,3)=imresize(im4Bis(:,:,3),[256 256]);
figure(1); imshow(im4);
figure(2); imshow(im10);
```

On peut aussi considérer une partie de l'image :

```
im9=im2(1:128,1:128);
figure(1); imshow(im2);
figure(2); imshow(im9);
```

Matlab dispose d'une aide en ligne sur la fenêtre de commandes : help permet d'afficher les différentes sections, help nom d'une section permet d'avoir la liste des commandes dans cette section, en particulier help images donne la liste des commandes relatives à l'image. help nom d'une commande donne une explication sur la commande et souvent un exemple qui peut vraiment être essayé. La liste des images disponibles sous Matlab est help imdemos.

On peut aussi réaliser une image en lui donnant des valeurs. Ainsi les commandes suivantes permettent de réaliser une image rouge et bleue.

```
im5=zeros(10,10,3);
im5(:,:,1)=[ones(10,5) zeros(10,5)];
im5(:,:,3)=[zeros(10,5) ones(10,5)];
figure(1); imshow(im5);
```

Autant il est en général difficile de colorer une image en niveaux de gris, il est simple de transformer une image en niveaux de gris.

```
im6=rgb2gray(im4);
figure(1); imshow(im4);
figure(2); imshow(im6);
```

Les fonctions Matlab pour lire et enregistrer les images sont imread et imwrite. Les fonctions disponibles pour afficher les images sont image, images et imshow. Attention à imshow qui peut étendre la palette de couleur ou de niveau de gris sans prévenir. Plusieurs images peuvent être affichées sur une seule figure grâce aux fonctions figure et subplot. Une image en niveau de gris ou n'importe quelle matrice peut être considérée comme une surface : à chaque coefficient de la matrice, on associe un point dont l'abscisse et l'ordonnée sont déterminés par la position de ce coefficient et la cote est déterminée par la valeur de ce coefficient. Les fonctions Matlab disponibles sont surf et mesh, la première fonction colore les éléments de surface tandis que la deuxième colore seulement les bords des éléments de surface.

En ce qui concerne l'interface Matlab et la disposition des fenêtres, si celle-ci n'est pas satisfaisante, il est possible de revenir à la disposition par défaut en actionnant successivement dans les menus déroulant : Desktop, Desktop Layout, Default.

Exercices

Exercice 1 (17)

Cet exercice illustre la commande ind2rgb exposée dans le cours 1 (p. 2). On considère une image couleur définie par une table de chiffres et une table de couleurs ; en fait chaque chiffre de la première table correspond à une couleur définie par la deuxième table.

$$[g_{mn}] = \begin{bmatrix} 1 & 1 & 2 & 2 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 2 & 4 & 5 & 6 & 7 \\ 1 & 1 & 2 & 2 & 4 & 5 & 6 & 7 \\ 2 & 2 & 2 & 2 & 5 & 5 & 6 & 7 \\ 2 & 2 & 2 & 2 & 4 & 8 & 7 & 8 \\ 4 & 4 & 4 & 4 & 3 & 3 & 7 & 8 \\ 5 & 5 & 5 & 2 & 5 & 5 & 8 & 8 \\ 6 & 6 & 5 & 2 & 6 & 6 & 6 & 7 \end{bmatrix}$$

$$(1)$$

la table de couleur est définie par :

$$t = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$
 (2)

La première colonne est associée à la couleur rouge, la deuxième colonne est associée à la couleur verte, la troisième colonne est associée à la couleur bleue.

- 1. Ecrivez les trois matrices R, G, B correspondant à cette image couleur.
- 2. Donnez un nom à chacune des couleurs présentes dans cette image.
- 3. On forme un image en niveau de gris contenant la composante rouge. Ecrivez cette matrice.
- 4. Formez les trois matrices R,G,B en supprimant la composante rouge.

Travaux pratiques

Préambule

(2)

Travaux Pratiques 1 Pour se familiariser avec Matlab, faire les deux traitements suivants

1. Faire les deux premières questions de l'exercice 1 en utilisant la commande Matlab ind2rgb. Il est nécessaire de recopier t. Pour G, on peut le synthétiser de façon aléatoire avec la commande suivante.

```
G=ceil(7*rand(8,8));
```

- 2. choisir une image en niveau de gris, une image parmi les images présentes dans Matlab, imdemos. Cette image doit être en 256 niveaux de gris (par exemple en appliquant à une image colorée l'instruction Matlab rgb2gray) et de taille 256 × 256 (par exemple en considérant qu'une partie de l'image). On note cette image image_n.
- 3. Construire une image synthétique : soit un carré blanc noyé dans un fond uniforme gris, soit un disque noyé dans un fond uniforme. La première image peut se faire avec im=ones(256,256) *0.5; pour le fond gris et avec im(128-15:128+15,128-15:128+15)=1; pour le carré blanc à l'intérieur. La deuxième image peut se faire avec meshgrid et find en utilisant le fait qu'un disque a pour équation : $(x-100)^2 + (y-100)^2 \le 400$.

```
[y,x] = meshgrid(0:255,0:255);
```

L'image synthétique choisie est notée image_s. Que faut-il changer dans la commande pour que le disque soit centré ? Que faut-il changer dans la commande pour que le disque soit plus grand ?

- 4. Choisir une image colorée que l'on notera image_c, afficher en rouge la composante rouge de l'image, commenter. Puis afficher en vert la composante verte de l'image. Et enfin afficher en bleu la composante bleue de l'image. Commenter.
- 5. Une façon de fabriquer une texture déterministe consiste à répliquer un motif de taille 8×8 sur une image. Le motif est tiré aléatoirement par exemple suivant une loi uniforme sur [0,1].

```
motif=rand(8,8);
```

On pourra utiliser la fonction repmat.

Exercices

Exercice 2 (2) On cherche à définir deux images décrites par deux suites $g_{m,n}^a$ et g_{mn}^b représentant l'une un carré centré de taille 30×30 , l'autre un disque centré de diamètre 30 pixels au sein d'une image 256×256 .

- 1. On se place tout d'abord dans un plan à deux dimensions. On note un point M de coordonnées (x,y). Définir un ensemble de conditions sur les coordonnées pour que M soit dans un carré de taille 30×30 , centré en O' de coordonnées (128,128).
- 2. En déduire une formule pour la suite $g_{m,n}^a$ qui vaut 0.4 en dehors du carré et 1 dans le carré au moyen de la fonction caractéristique notée $\mathbf{1}$ ($\mathbf{1}_A(m,n)=1$ quand $(m,n)\in A$ et $\mathbf{1}_A(m,n)=0$ sinon. Pour un signal à temps discret, $g_n=0.4+0.6\mathbf{1}_{\{121..136\}}[n]$ est une suite qui vaut 0.4 partout sauf pour les indices entre 121 et 136 où elle vaut 1.
- 3. De nouveau on se place dans un plan à deux dimensions. On note un point M de coordonnées (x,y) et O' le centre de l'image de coordonnées (128,128). Définir une condition sur M puis sur les coordonnées pour que M soit dans le disque de diamètre 30.

4. En déduire une formule pour la suite $g_{m,n}^b$ qui vaut 0.4 en dehors du disque et 1 dans le disque au moyen de la fonction 1.

Exercice 3 (22) On considère une image g_{mn} de taille $M \times N$, en niveaux de gris à valeurs sur [0,1]. Comment cette image est-elle modifiée par les transformations suivantes sur les niveaux de gris :

- 1. $f_{mn} = 1 g_{mn}$
- 2. $f_{mn} = 0$ si $g_{mn} < 0.5$ et $f_{mn} = 1$ si $g_{mn} \ge 0.5$.
- 3. $f_{mn} = g_{M-m,n}$.

Travaux pratiques

Travaux Pratiques 2 (43)

En utilisant les commandes Matlab fliplr et éventuellement find expérimentez les transformations présentées dans l'exercice 3 sur une image en niveau de gris.

1 Analyse du signal image

(45)

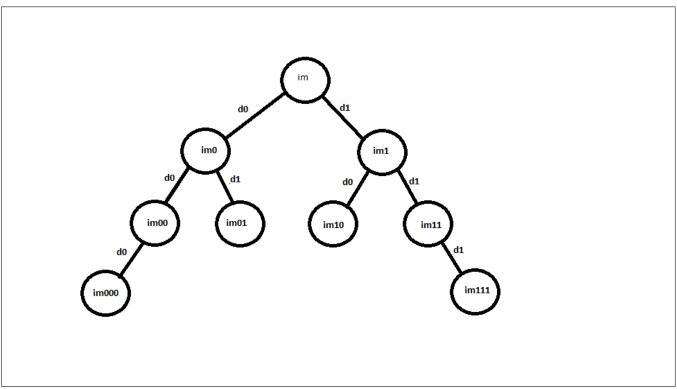


Figure 1: Schéma pour les sous-échantillonnages successifs du TP3 ; im désigne soit l'image synthétique choisie soit l'image naturelle choisie ; d0 et d1 désignent respectivement deci0 et deci1.

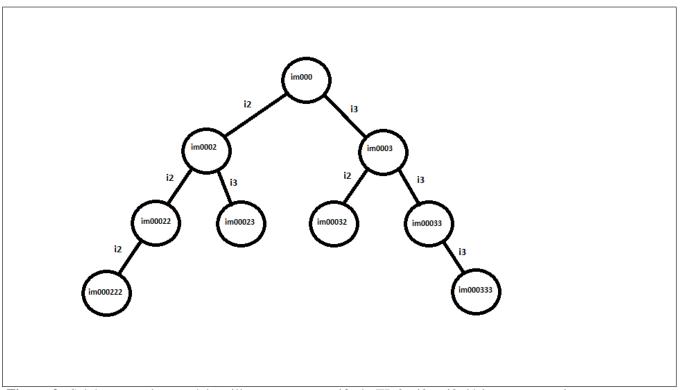


Figure 2: Schéma pour les sur-échantillonnages successifs du TP 3 ; i2 et i3 désignent respectivement inte2 et inte3.

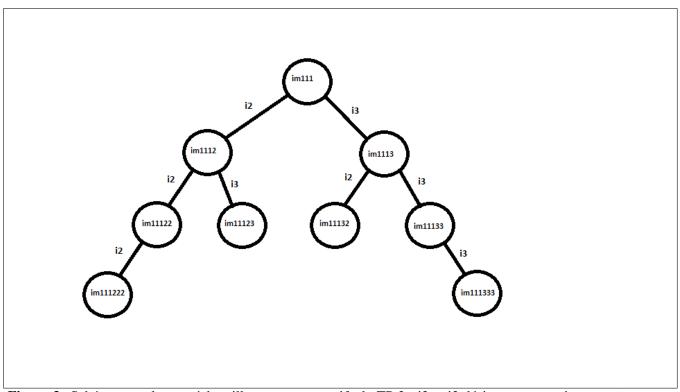


Figure 3: Schéma pour les sur-échantillonnages successifs du TP 3 ; i2 et i3 désignent respectivement inte2 et inte3.

Travaux Pratiques 3 On cherche ici à sous-échantillonner une image puis à partir de l'image sous-échantillonnée à retrouver l'image de départ en sur-échantillonnant l'image de taille réduite; ceci en s'inspirant de ce qui a été vu en traitement du signal. Pour sous-échantillonner la première technique, notée deci0 consiste à prélever une ligne sur deux et une colonne sur deux (c'est-à-dire un pixel sur quatre), la deuxième technique, notée deci1

consiste à appliquer une filtre passe-bas avant de faire le prélèvement. Ces deux techniques transforment une image de taille $M \times N$ en une image de taille $\frac{M}{2} \times \frac{N}{2}$.

Pour sur-échantillonner, la première technique, notée inte2, consiste à répéter chaque ligne et chaque colonne, c'est-à-dire que chaque pixel se trouve entouré de trois nouveaux pixels qui ont la même valeur que ce pixel. La deuxième technique inte3 consiste à appliquer un filtre passe-bas après avoir le rajout des pixels supplémentaires, ce faisant les pixels rajoutés auront des valeurs intérmédiaires entre les pixels qui les entourent. Ces deux dernières techniques appliquées à une image de taille $M' \times N'$ en une image de taille $2M' \times 2N'$.

L'objectif est d'appliquer l'une des deux premières techniques deci0 ou deci1 trois fois de suite de façon à ce que le nombre de pixels de l'image soit divisé par 64, figure 1. Parmi les huit images que l'on pourrait obtenir, on considère seulement les images notées im000 et im111. Ensuite on applique à ces deux (im000 et im111) l'une des deux autres techniques inte2 et inte3 de façon à retrouver pour les différentes images la taille initiale. Lorsqu'on considère initialement l'image im000, les images obtenues avec le schéma de la figure 2 sont im000222 et im000333. Lorsqu'on considère initialement l'image im000, les images obtenues avec le schéma de la figure 3 sont im111222 et im111333.

Ces deux schémas de sous-échantillonnage sont appliqués d'abord à une image synthétique notée image_s puis à une image naturelle image_n. La première image est obtenue en échantillonnant l'image définie par $g(x,y)=0.5+0.5\cos(2\pi(3x+4y))$ et avec $\Delta x=\Delta y=0.05$ en s'assurant qu'elle forme une image de taille 256×256 . La deuxième image est une des images provenant de imdemos, transformée de façon à ce qu'elle soit en niveaux de gris et de taille 256×256 . Le résultat est noté image_n.

Du point de vue de l'implémentation Matlab, les transformations deci0, deci1, inte2, inte3 peuvent s'implémenter sous la forme de fonctions en ligne.

```
deci0=@ (im) im (1:2:end, 1:2:end);
deci1=@ (im) deci0 (filter2 (ones (2) /4, im));
inte2=@ (im) im ([1:end; 1:end], [1:end; 1:end]);
inte3=@ (im) filter2 (ones (2) /4, inte2 (im));
```

On peut représenter le spectre d'une image par les commandes suivantes :

```
im2=imread('tire.tif');
figure(1); imshow(im2);
sp2=fftshift(fft2(((im2))))/prod(size(im2));
figure(2); imshow(mat2gray(20*log10(1e-6+abs(sp2))));
```

- 1. Construire image_s. Discuter du critère de Nyquist sur cette image.
- 2. Construire im000 à partir de image_s.
- 3. Construire im111 à partir de image_s.
- 4. Construire im101 à partir de image_s.
- 5. Construire im000222 à partir de image_s.
- 6. Construire im000333 à partir de image_s.
- 7. Construire im111222 à partir de image_s.
- 8. Construire im111333 à partir de image_s.
- 9. Construire le spectre associé à im111333 et comparer le au spectre de image_s.
- Construire le spectre associé à im000222 et comparer le au spectre de image_s.
- 11. Construire image_n.
- 12. Construire im000 à partir de image_n.

- 13. Construire im111 à partir de image_n.
- 14. Construire im101 à partir de image_n.
- 15. Construire im000222 à partir de image_n.
- 16. Construire im000333 à partir de image_n.
- 17. Construire im111222 à partir de image_n.
- 18. Construire im111333 à partir de image_n.
- 19. Construire le spectre associé à im111333 et comparer le au spectre de image_n.
- 20. Construire le spectre associé à im000222 et comparer le au spectre de image_n.

Exercices

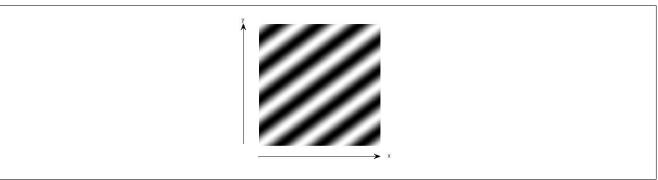


Figure 4: image de l'exercice 4

Exercice 4 (5) On analyse les problèmes de sous-échantillonnage, repliement de spectre et préfiltrage à partir du signal $f(x,y) = 0.5 + 0.5\cos(2\pi(3x+4y))$ échantillonnée $\Delta x = \Delta y = 0.2$. La figure 4 montre cette image non-échantillonnée.

- 1. Que vaut le signal échantillonnée g_{mn} lorsqu'on se restreint à 256×256 points ?
- 2. Quelles est le spectre de f (i.e. avant échantillonnage) ? Montrez que le spectre n'est pas F(u, v) = 0.5δ(u, v) + 0.25δ(u 3) + 0.25δ(u + 3) + 0.25δ(v 4) + 0.25δ(v + 4) en calculant la transformée de Fourier inverse. Comme f est périodique, le spectre de f est constitué de pics. Proposer une solution et vérifier cette solution en utilisant la transformée de Fourier inverse. Représentez où se trouvent ces pics à partir de leur coordonnées en (u, v) ∈ [-5, 5].
- 3. Le critère de Nyquist est donné ici par $|u| \le \frac{1}{2\Delta x}$ et $|v| \le \frac{1}{2\Delta y}$. Tracer le domaine correspondant à ce critère de Nyquist. Est-ce que ce critère est vérifié ?
- 4. Donnez l'expression du spectre de l'image échantillonnée Fe(u,v) et représentez le résultat. Pour cela il suffit de tracer les symétries centrales en les quatre extrémités du critère de Nyquist. Justifiez cette construction géométrique.
- 5. Pour retrouver l'image continue à partir de l'image échantillonnée, il suffit normalement de filtrer l'image avant échantillonnage par un filtre dont la réponse en fréquence est $H(u,v) = \mathbf{1}_{[-2.5,2.5]}(u)\mathbf{1}_{[-2.5,2.5]}(v)$. Expliquez d'où vient le nombre 2.5 ? Quels sont les pics du spectre inital qui sont conservés ?
- 6. Calculez l'expression du spectre de l'image filtrée avant échantillonnage $F_h(u,v)$.
- 7. En déduire l'image filtrée avant échantillonnage $f_h(x, y)$.

Cours 2 (4)

Les courbes de niveaux (ou lignes de niveaux) sont utilisées sur les cartes géographiques et indiquent les points de la carte de même altitude. Si on se déplaçait le long de ces lignes, on ne monterait pas et on ne descenderait pas non plus. Les canaux, les bords de lacs, les plages le long de la mer sont sur des lignes de niveaux. On peut utiliser la notion de courbes de niveaux pour aider à la visualisation de surface dans un espace à trois dimensions. Faire une courbe de niveau, cela consiste à dessiner la section d'une surface suivant un plan horizontal qui se situe à une hauteur z. Ainsi les courbes de niveaux d'une sphère sont des cercles concentriques autour d'un point qui est à la verticale du centre de la sphère. Mathématiquement, on définit, ici, la surface comme une fonction qui à chaque point du plan horizontal (x,y) associe un point sur la surface et à la verticale de ce point du plan horizontal (x,y,g(x,y)). On peut alors définir une courbe de niveau comme étant l'ensemble des points du plan horizontal (x,y) pour lesquels les points sur la surface associés sont à la cote z. Ainsi une courbe de niveau est donnée par $\{(x,y)|g(x,y)=z\}$. Il n'y a pas vraiment de règles pour définir les différentes cotes z_i associées à chacune des courbes de niveaux. Le choix doit être fait de façon à aider à la visualisation de la surface.

Exercice 5 (4)

On cherche à représenter le signal $f(x,y) = 0.5 + 0.5\cos(2\pi(3x+4y))$ pour $x,y \in [0,1]$ et à étudier sa périodicité.

- 1. Réprésentez en rouge l'ensemble des points (x, y) tels que f(x, y) = 1. Ce sont les crêtes de la surface, ce sont les points qui apparaitraient en blanc sur l'image de f.
- 2. Représentez en vert l'ensemble des points (x, y) tels que f(x, y) = 0. Ce sont les vallées de la surface, ce sont les points qui apparaîtraient en noir sur l'image de f.
- 3. Quelle est la période de $x \mapsto f(x,y_0)$? On note cette valeur sous forme d'un vecteur $t_x = (T_x,0)$. Montrez sur le graphique que si un point M=(x,y) est sur une courbe rouge, alors $M+t_x=(x+T_x,y)$ est encore sur une courbe rouge. Quelle est la période de $y\mapsto f(x_0,y)$? On note cette valeur sous forme d'un vecteur $t_y=(0,T_y)$. Montrez ce qui se passe alors sur le graphique. Quelle est la périodicité de $\lambda\mapsto f(x_0+\lambda,y_0+\lambda)$? On note cette valeur sous forme d'un vecteur $t_{xy}=(T_{xy},T_{xy})$. Montrez ce qui se passe alors sur le graphique.
- 4. On définit la périodicité pour f par un vecteur de translation qui est orthogonale aux crêtes et aux vallées. Tracez graphiquement ce vecteur, quelles sont les coordonnèes de ce vecteur.
- 5. Vérifiez qu'effectivement f est invariante par translation t = (3/25, 4/25) (ce vecteur est orthogonal puisque coolinéaire à (3,4) qui est normale à toutes les crêtes).

Cours 3 (5) La notion d'images naturelles et d'images synthétiques est présentée dans le TP 1. Pour l'image synthétique on pourra ici choisir celle produite par les commandes suivantes :

```
[J,I]=meshgrid((0:255)/255,(0:255)/255);
image_s=2*I.*(I<=0.5)+2*(1-I).*(I>0.5);
```

2 Quantification

Travaux Pratiques 4 (5) On cherche à appliquer une quantification uniforme sur une image en niveaux de gris.

- 1. Afficher l'image synthétitique en niveaux de gris choisie.
- 2. Vérifier (ou s'assurer) que cette image est définie par une matrice 2D avec des valeurs entre 0 et 1. L'image ainsi obtenue est notée image_s.
- 3. Appliquer une quantification sur 30 niveaux en s'inspirant de ce que

```
imQ=floor(30*im)/30;
```

L'image obtenue doit ressembler à l'image initiale.

- 4. Appliquer une quantification sur 10 niveaux.
- 5. Appliquer une quantification sur 5 niveaux.
- 6. Afficher l'image naturelle en niveaux de gris choisie
- 7. Vérifier (ou s'assurer) que cette image est définie par une matrice 2D avec des valeurs entre 0 et 1. L'image ainsi obtenue est notée image_n.
- 8. Appliquer une quantification sur 30 niveaux. L'image obtenue doit ressembler à l'image initiale.
- 9. Appliquer une quantification sur 10 niveaux.
- 10. Appliquer une quantification sur 5 niveaux.
- 11. Si on appliquait cette méthode de quantification à des images couleurs, cela reviendrait à quantifier les composantes couleurs et non les couleurs elles-mêmes. Expliquer la différence entre les deux notions.

3 Histogramme

Cours 4 (4) Pour une image en niveaux de gris, on appelle histogramme le fait de représenter le nombre (ou la proportion) de pixels ayant tels niveaux de gris en fonction du niveau de gris. Sous Matlab, les commandes suivantes permettent de tracer un histogramme.

```
[y,x]=hist(im(:),50);
figure(1); plot(x,y);
```

où 50 est ici le nombre de plages de niveaux de gris considérés.

Travaux Pratiques 5 (4) On cherche à représenter l'histogramme de différentes images en niveaux de gris.

- 1. Afficher sur un graphe l'histogramme sur 30 niveaux de image_s défini lors du TP 4, p. 12. En observant d'une part image_s et d'autre part l'histogramme, indiquer à quelle partie de l'image correspond tel pic de l'histogramme.
- 2. Afficher sur un graphe l'histogramme sur 10 niveaux de image_s. Montrer le lien qu'il y a entre l'histogramme sur 10 niveaux et l'histogramme sur 30 niveaux.
- 3. Afficher sur un graphe l'histogramme sur 5 niveaux de image_s. Montrer le lien qu'il y a entre l'histogramme sur 5 niveaux et l'histogramme sur 10 niveaux.
- 4. Afficher sur un graphe l'histogramme sur 30 niveaux de image_n défini lors du TP 4, p. 12. En observant d'une part image_n et d'autre part l'histogramme, indiquer à quelle partie de l'image correspond tel pic de l'histogramme.
- 5. Afficher sur un graphe l'histogramme sur 10 niveaux de image_n. Montrer le lien qu'il y a entre l'histogramme sur 10 niveaux et l'histogramme sur 30 niveaux.
- 6. Afficher sur un graphe l'histogramme sur 5 niveaux de image_n. Montrer le lien qu'il y a entre l'histogramme sur 5 niveaux et l'histogramme sur 10 niveaux.
- 7. Supposons maintenant que par inadvertance on ait appliqué l'instruction présentée lors du cours 4 sur une image couleur, on obtiendrait évidemment pas exactement un histogramme, mais en fait une superposition de trois histogrammes. Expliquer plus précisément ce qui se passe.

Exercices

Exercice 6 (3) On cherche à calculer à la main un histogramme sur trois niveaux de l'image $g_{mn} = \frac{m}{10} \mathbf{1}_{0..9}[m] + (2 - \frac{m}{10}) \mathbf{1}_{10..19}[m]$ L'histogramme peut se voir aussi comme une quantification des niveaux de gris de l'image et ensuite on compte le nombre de pixel associé à chaque niveau de quantification et on les représente sur un graphique. Pour simplifier le calcul l'image étudiée est invariante par translation suivant la première coordonnée. Ainsi l'histogramme de l'image est en fait identique (presque) à l'histogramme d'un profil horizontal.

- 1. Représentez le signal $g(x) = 2x\mathbf{1}_{[0,1/2]}(x) + (2-2x)\mathbf{1}_{[1/2,1]}(x)$ sur [0,1].
- 2. Représentez sur la même figure la quantification sur trois niveaux régulièrement répartis du signal $g_q(x) = Q(g(x))$. Pour quantifier un signal sur trois niveaux, on définit une partition de l'intervalle [0,1] sur l'axe des ordonnées en trois intervalles I_k de même taille, on affecte à chaque niveau de gris g le milieu m_k de l'intevalle I_k auquel g appartient : $Q(g) = m_k$ si $g \in I_k$.
- 3. L'image est en fait de taille 20×20 , représentez sur la même figure le profil de l'image et de l'image quantifiée. Cela revient à échantillonner g(x) et $g_q(x)$ avec une période d'échantillonnage de $\frac{1}{20}$.
- 4. Tracez l'histogramme de l'image sur 3 niveaux. Pour cela on comptabilise le nombre de pixels sur chaque niveaux de quantification. L'histogramme est le nombre de pixel en fonction du niveau de quantification.

4 Fréquences associées à une image

Travaux pratiques

Cours 5 (30) La transformée de Fourier adaptée aux images est la transformée de Fourier discrète bidimensionnelle :

$$G_{k,l} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{m,n} e^{-j2\pi(\frac{km}{M} + \frac{ln}{N})}$$

où $g_{m,n}$ est l'intensité (valeur entre 0 et 1) du pixel à la position (m,n) et $G_{k,l}$ est le coefficient associé à la fréquence dont la composante horizontale u est $\frac{k}{M}f_e$ et la composante verticale v est $\frac{l}{N}f_e$, f_e étant la fréquence

d'échantillonnage qui s'exprime dans une unité. Les variables associées aux fréquences spatiales sont notées u et v. La taille de l'image est $M \times N$. Le choix du coefficient $\frac{1}{N^2}$ est dans une certaine mesure arbitraire, il garantit ici que la composante constante $G_{0,0}$ est la moyenne du signal image.

La transformée de Fourier discrète ne tient pas compte du fait que les images sont à valeurs réelles et que par suite elle sont en plus d'être périodiques, symétriques par rapport à la moitié de la fréquence d'échantillonnage. Il faut donc comme en traitement du signal utiliser l'instruction fftshift pour placer la fréquence nulle au centre de l'image et ainsi mieux visualiser les propriétés de symétrie.

La transformée de Fourier discrète inverse est :

$$g_{m,n} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G_{k,l} e^{j2\pi(\frac{km}{M} + \frac{ln}{N})}$$

Cette formule exprime le fait que les images sont des combinaisons linéaires d'images formées de sinusoïdes, il suffit pour cela d'ajouter les exponentielles complexes qui sont conjuguées avec des coefficients qui sont conjugués $(G_{k,l} = \overline{G_{M-k,N-l}} \text{ si } k \neq 0 \text{ et } l \neq 0).$

Travaux Pratiques 6 Les instructions suivantes montrent comment construire ces différentes sinusoïdes qui permettent par approximation successive de reconstruire l'image d'origine.

```
im=zeros(128); im(64-9:64+9,64-9:64+9)=1;
sp=fft2(im);
sp1=zeros(128);
sp1(1,1)=sp(1,1);
sp1(2,1)=sp(2,1);
sp1(128,1)=sp(128,1);
sp1(3,1)=sp(3,1);
sp1(127,1)=sp(127,1);
im1=ifft2(sp1);
figure(1); imshow([im im1]);
```

Ces instructions permettent de transformer un carré im en un fond noir avec une bande horizontale légèrement plus claire.

- 1. Modifiez ces instructions de façon à obtenir une bande verticale.
- 2. Modifiez ces instructions de façon à obtenir une bande horizontale ainsi qu'une bande verticale.
- 3. Appliquez ces différentes instructions à image_n définie dans le TP 4, p. 12 de façon à obtenir trois images.

L'animation suivante permet de rendre plus visible le fait les différentes sinusoïdes permettent par approximation successive de reconstruire l'image d'origine.

```
im=zeros(128); im(64-9:64+9,64-9:64+9)=1;
points=zeros(128);
sp=fftshift(fft2(im));
sp1=zeros(128);
im1=real(ifft2(fftshift(sp1)));
figure(1); imshow([im im1 points]);
for k=1:1000
   [J,I]=ginput(1);
   I=round(I); J=round(J); J=J-floor((J-1)/128)*128;
   if (J<=127)&(J>=2)&(I<=127)&(I>=2)
        sp1(I-1:I+1,J-1:J+1)=sp(I-1:I+1,J-1:J+1); points(I-1:I+1,J-1:J+1)=1;
```

```
end;
im1=mat2gray(real(ifft2(fftshift(sp1))));
figure(1); imshow([im im1 points]);
end;
```

- 4. Decrire comment on peut retrouver une bonne approximation du carré à l'aide de cette animation. Afficher l'image obtenue avec cette animation.
- 5. Expliquer comment le spectre contient les informations nécessaires pour reconstruire une image à partir de différentes sinusoïdes.

Cours 6 De même qu'en traitement du signal, on visualise le spectre en mettant la module sur l'axe des ordonnées soit avec une échelle linéaire soit avec une échelle logarithmique. L'échelle linéaire rend compte effectivement de la proportion avec laquelle la forme de l'image résulte de la sinusoïde. Souvent l'essentiel le spectre a une valeur en la fréquence nulle largement plus importantes qu'en les autres fréquences. L'échelle logarithmique permet de visualiser avec précisions la valeur du module du spectre en les fréquences où ce module est très faible. Cette échelle logarithmique peut aussi conduire à des erreurs d'interprétations.

On peut représenter le spectre de la transformée de Fourier en perspective. Mais dans la mesure où il s'agit de la transformée de Fourier d'une grande image, c'est souvent très difficile à visualiser. Une autre façon de procéder est de visualiser le profil d'un spectre (i.e. une ligne ou une colonne de la matrice correspondant au module du spectre). Une autre façon est de représenter le module du spectre sous la forme d'une image dont les points blancs correspondent aux valeurs les plus élevés de ce module du spectre, tandis que les points noirs correspondent au points les moins élevés. Cette représentation est souvent faite en utilisant une échelle logarithmique pour le module du spectre.

De manière analogue au traitement du signal, on pourrait fabriquer une échelle indiquant les fréquences spatiales en cm⁻¹ à partir de la seule connaissance de la taille de l'image $10cm \times 10cm$. L'habitude en traitement d'image est d'exprimer plutôt les fréquences spatiales soit en nombre de cycles par image, soit en nombre de cycles par degré. La deuxième unité dépend de la taille de l'image et la troisième unité dépend de l'angle sous lequel l'image est vue. Chaque fréquence spatiale correspond à une sinusoïde qui a un certain nombre de cycles dans l'image. La deuxième unité dépend de l'angle sous lequel l'image est visualisée par un observateur.

Travaux Pratiques 7 Pour chacune des questions suivantes choisissez un mode de représentation qui vous semblera le plus adapté de manière à montrer les éléments suivants :

- 1. transformée de Fourier d'un carré blanc sur fond sombre,
- 2. transformée de Fourier d'un disque blanc sur fond sombre,
- 3. transformée de Fourier d'un ensemble de raies blanches sur fond sombre,
- 4. transformée de Fourier d'une sinusoïde,
- 5. propriété d'invariance du module par translation d'une image,
- 6. propriété d'invariance par rotation,
- 7. transformée de Fourier d'une texture déterministe (i.e. motif de petite taille répétée un grand nombre de fois).

Exercices

Cours 7 (6)

La transformée de Fourier adaptée aux images est la transformée de Fourier discrète bidimensionnelle :

$$G_{k,l} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{m,n} e^{-j2\pi(\frac{km}{M} + \frac{ln}{N})}$$

où $g_{m,n}$ est l'intensité (valeur entre 0 et 1) du pixel à la position (m,n) et $G_{k,l}$ est le coefficient associé à la fréquence dont la composante horizontale u est $\frac{k}{N}f_e$ et la composante verticale v est $\frac{l}{N}f_e$, f_e étant la fréquence d'échantillonnage qui s'exprime dans une unité. Les variables associées aux fréquences spatiales sont notées u et v. La taille de l'image est v est v le choix du coefficient v est dans une certaine mesure arbitraire, il garantit ici que la composante constante v est la moyenne du signal image.

Exercice 7 (6)

On considère une image de taille 4×4 définie par

$$g_{m,n} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
 (3)

- 1. Montrez que les coefficients de la transformée de Fourier discrète bidimensionnelle sont la somme de deux termes, chacun correspondant à une exponentielle complexe.
- 2. Calculez le module et l'argument de chacun de ces coefficients.
- 3. Calculez les fréquences spatiales de chaque coefficient suivant les deux types d'échelles fréquentielles (celle en fréquences réduites et provenant du traitement du signal et celle en cycles par image), on pourra présenter ces fréquences spatiales sous la forme de deux matrices $[u_{k,l}]$ et $[v_{k,l}]$.

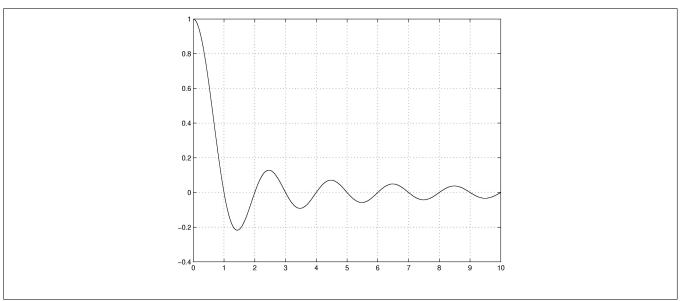


Figure 5: $u \mapsto \frac{\sin(\pi u)}{\pi u}$, (exercice 8), (10)

Cours 8 (10)

1. La transformée sur des signaux à temps continu est définie par

$$\mathcal{T}F[g(x)](u) = \int_{-\infty}^{+\infty} g(x)e^{-j2\pi ux} dx \tag{4}$$

2. La transformée de Fourier d'une fonction constante sur un intervalle symétrique est donnée par :

$$\mathcal{T}F[\mathbf{1}_{[-\frac{1}{2},\frac{1}{2}]}(x)](u) = \frac{\sin(\pi u)}{\pi u}$$
(5)

Cette fonction est représentée sur la figure 5 (p. 16).

Le calcul se fait en restreignant les bornes de l'intégrale (4), en calculant une primitive de $x \mapsto e^{-j2\pi ux}$ et enfin en reconnaissant dans la différence entre cette primitive appliquée en $x = \frac{1}{2}$ et en $x = -\frac{1}{2}$, l'expression donnée par (5).

3. La transformée de Fourier d'un dirac est une exponentielle complexe.

$$\mathcal{T}F[\delta(x - x_0, y - y_0)](u, v) = e^{-j2\pi(ux_0 + vy_0)}$$
(6)

4. La transformée de Fourier d'une fonction gaussienne est donnée par :

$$\mathcal{T}F[e^{-x^2}](u) = \sqrt{\pi}e^{-\pi^2 u^2} \tag{7}$$

On commence d'abord par calculer le résultat en la fréquence nulle, pour cela on calcule $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-x^2-y^2} dxdy$ que l'on peut approcher par en passant en coordonnée polaire $\int_{0}^{+\infty} \int_{0}^{2\pi} e^{-\rho^2} \rho d\rho d\theta$ ce qui vaut π et par suite $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$. Ensuite en regroupant les termes exponentiels qui apparaissent dans $TF[e^{-x^2}](u)$, on voit que $x^2 + j2\pi ux$ forme le début d'un terme carré $(x+j\pi ux)^2$ auquel il faut rajouter $\pi^2 u^2$, ce dernier terme ne dépend pas de x et peut donc sortir de l'intégrale. On a ainsi montré que $TF[e^{-x^2}](u) = e^{-\pi^2 u^2} \int_{-\infty}^{+\infty} e^{-(x+j\pi ux)^2} dx$. La fonction $z \mapsto e^{-z^2}$ est holomorphe sur une partie du plan complexe, aussi l'intégrale ne dépend pas de u et vaut donc $\sqrt{\pi}$.

5. La transformée de Fourier vérifie une propriété utilisée pour la modulation :

$$\mathcal{T}F[g(x)e^{-j2\pi u_0 x}](u) = \mathcal{T}F[g(x)](u - u_0)$$
(8)

Elle se démontre en explicitant le premier membre et en regroupant les exponentielles.

6. Un retard sur le signal se traduit du point de vue de la transformée de Fourier par un déphasage.

$$\mathcal{T}F[g(x-x_0)](u) = \mathcal{T}F[g(x)](u)e^{-j2\pi ux_0}$$
 (9)

En effet $g(x-x_0)$ s'exprime comme la transformée de Fourier inverse de $\mathcal{T}F[g(x)](u)$ appliqué en $x-x_0$, en scindant l'exponentielle, on observe que $g(x-x_0)$ est aussi la transformée de Fourier inverse de $\mathcal{T}F[g(x)](u)e^{-j2\pi ux_0}$ appliqué en x.

7. Une rotation d'une fonction à deux variable se traduit du point de vue la transformée de Fourier par une rotation identique des fréquences spatiales. Ainsi si (X,Y) sont les coordonnées d'un nouveau point obtenu par rotation d'angle θ à partir d'un point (x,y) alors la transformée de Fourier de g̃(x,y) = g(X,Y) est donnée par TF[g̃(x,y)](u,v) = TF[g(X,Y)](U,V). Cette invariance par rotation de la tranformée de Fourier signifie seulement que si deux observateurs regardent le même phénomène en deux endroits différents, ils observent le même phénomène, même si ce phénomène est décrit de manière différente en coordonnée cartésienne. La rotation est définie par

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$
 (10)

La démonstration provient de ce que le produit scalaire entre le vecteur (x,y) et (u,v) qui intervient dans la définition de la transformée de Fourier est égale au produit scalaire entre (X,Y) et (U,V) du fait que la matrice de rotation est dite orthogonale $(D^TD=1)$.

Ce résultat n'est absolument pas valable pour des transformations qui ne conserveraient pas les distances par exemple pour des compositions d'homothéties et de rotations.

Exercice 8 (10)

On cherche à calculer et à visualiser des transformées de Fourier d'images définies par des fonctions de deux variables particulières.

- 1. Calculez et dessinez à main levée la transformée de Fourier de $\mathbf{1}_{[-L,L]}(x)$, avec L=1 et $u\in[-2,2]$. Exprimez en fonction de L la largeur du premier lobe et des lobes suivants.
- 2. Calculez la transformée de Fourier de $\mathbf{1}_{[-L,L]}(x)\mathbf{1}_{[-3L,3L]}(y)$. Représentez avec $L=1, u,v\in [-1,1]$, les droites où le module de cette transformée de Fourier s'annule. Ces droites se coupent à angle droit et forment des rectangles qui délimitent les lobes, mettre une croix au sein de ces lobes pour indiquer le maximum et mettre la valeur du module en ce maximum. Exprimez en fonction de L des grandeurs caractéristiques des différents lobes.
- 3. Représentez l'image définie par

$$\mathbf{1}_{[-L,L]}(x)\mathbf{1}_{[-3L,3L]}(y)$$

dans un repère en trois dimensions représentée en perspective, puis sous la forme d'une image. Représentez sous la forme d'une image, l'image définie par

$$\mathbf{1}_{[-L,L]}(x+y)\mathbf{1}_{[-3L,3L]}(x-y)$$

avec $L=1, x,y\in [-5,5]$. Quelles transformations géométriques permettent de transformer la première image en la deuxième, montrez qu'il ne s'agit pas d'une rotation, quelles sont les paramètres de cette transformation géométrique? En déduire le module de la transformée de Fourier de $\mathbf{1}_{[-L,L]}(x+y)\mathbf{1}_{[-3L,3L]}(x-y)$. Représentez ce module de spectre de la même façon que dans la question précédente.

- 4. Calculez et dessinez à main levée $\mathbf{1}_{[-L,L]}(x)(0.5+0.5\sin(2\pi fx))$ avec L=1, f=3 et $x\in[-2,2]$ puis le module de sa transformée de Fourier avec L=1, f=3 et $u\in[-5,5]$ en indiquant en particulier où ce module s'annule et de façon approximative les maxima en précisant pour ces maxima, la valeur de la phase.
- 5. Déduire de ce qui précède et par repliement de spectre, la transformée de Fourier du signal

$$f(x) = \mathbf{1}_{[-L,L]}(x)(0.5 + 0.5\sin(2\pi fx))$$

échantillonné à la fréquence fe=2f sur avec $L=1,\ f=3,\ u\in[-5,5]$. Quelle est la période d'échantillonnage? Donnez l'expression du signal échantillonné. Rajoutez le spectre du signal échantillonné sur le graphique précédent.

Exercice 9 (19) On considère une image f de taille 128×128 et une deuxième image g qui contient en son centre la première et tout autour que des zéros, sa taille est 256×256 . On a calculé la transformée de Fourier discrète G de g, comment retrouver la transformée de Fourier discrète F de f à partir de celle de G?

Exercice 10 (7) On s'intéresse au spectre d'une texture déterministe en suppposant qu'elle s'étend sur un objet suffisamment vaste. Une texture déterministe est souvent définie comme le fait de juxtaposer horizontalement et verticalement un même motif, un exemple est montré sur l'image de gauche de la figure 6 (p. 19).

- 1. On note x_n et X_k deux suites de même taille, telles que X_k soit la transformée de Fourier discrète de x_n . On créé une autre suite Y_k en intercallant des zéros sur les éléments X_k . Que devient alors y_n . On créé une autre suite z_n qui est la juxtaposition de x_n et x_n , que vaut Z_k .
- 2. On note g_{mn} une petite image et G_{kl} la transformée de Fourier discrète. Que vaut la transformée de Fourier discrète de la juxtaposition de g_{mn} avec g_{mn} .
- 3. Décrire une propriété des transformées de Fourier discrète des textures telles que définies dans le préambule.

Travaux pratiques

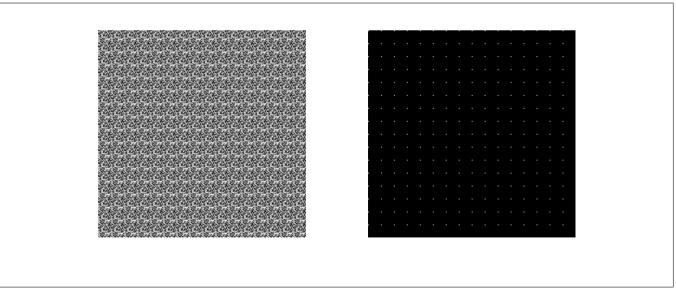


Figure 6: Exemple de texture déterministe et spectre de cette texture (exercice 10).

5 Spectres et directions privilégiées

(34)

Travaux Pratiques 8 On cherche à mettre en évidence la manière avec laquelle des directions privilégiées apparaissent sur le spectre d'une image en niveau de gris.

- 1. Récupérer l'image board.tif ou l'image blobs.png qui se trouvent dans le répertoire imdemos de Matlab. Transformer cette image en niveaux de gris en s'assurant que ses valeurs sont entre 0 et 1. Afficher l'image ainsi obtenue. C'est cette image que nous appelerons image_p.
- 2. Calculer le spectre de l'image avec l'instruction fft2. Puis appliquer l'instruction fftshift pour faire en sorte que les fréquences basses apparaissent au centre de l'image. Enfin appliquer abs pour avoir le module du spectre. Appliquer l'instruction

```
10*log10(...)
```

pour que les niveaux de gris affichés correspondent à une échelle logarithmique des valeurs du spectre. Appliquer l'instruction mat 2 gray de façon à convertir les valeurs ainsi obtenues en des valeurs entre 0 et 1. L'image obtenue doit présenter deux raies qui traversent l'ensemble de l'image.

3. On considère maintenant une autre image naturelle sans direction privilégiée et qui soit **en niveaux de gris** et à valeurs sur l'intervalle [0,1]. Appliquer l'instruction suivante

```
filter2 (ones (1,5) / sum (sum (ones (1,5))), image_n);
```

qui est un filtre directionnel. Afficher cette image qui est notée image_nd

- 4. Appliquer les mêmes instructions que pour image_p et discuter de l'apparition d'une direction privilégiée.
- 5. Appliquer l'instruction imrotate sur image_p de façon à obtenir trois images résultant d'une rotation avec plusieurs angles. Ces images sont notées image_pr1, image_pr2, image_pr3.
- 6. Appliquer encore ces mêmes instructions sur ces trois images et indiquer ce qu'il advient de ces directions privilégiées.

Travaux Pratiques 9 Pour mieux comprendre ce qui entraîne ces directions privilégiés on peut rechercher les pixels qui entraînent ces directions privilégiées. L'idée suggérée ici est de déformer le spectre uniquement en les points qui sont sur les raies qui témoignent de ces directions privilégiées et ensuite de visualier l'image ainsi transformée. Les zones de l'image qui sont le plus affectées par ces transformations sont justement celles qui sont le plus responsables des directions privilégiées.

```
im=double(rgb2gray(imread('board.tif')))/256;
spectre=fftshift(fft2(im));
index=find(mat2gray(20*log10(abs(spectre)))>=0.7);
spectre(index)=0;
ImTr=real(ifft2(fftshift(spectre)));
figure(1); imshow([im mat2gray(abs(im-ImTr))])
```

Travaux Pratiques 10 La fonction mat 2 gray peut avoir des conséquences surprenantes :

```
im1=ones(128);
im2=zeros(128); im2(10:50,10:50)=1;
im3=zeros(128); im3(23:26,23:26)=1;
ImA=(45-23-7)*im3+(23-7)*im2+7*im1;
ImB=(62-23-7)*im3+(23-7)*im2+7*im1;
figure(1); imshow([mat2gray(ImA) mat2gray(ImB); ImA/62 ImB/62]);
```

- 1. Parmi ces 4 images, quelles images laissent entendre qu'avec mat2gray on pourrait croire que ImA est similaire à ImB.
- 2. Parmi ces 4 images, quelles images montrent que ImA est bien différent de ImB.
- 3. expliquer ce qui dans le fonctionnement de mat2gray ce qui amène à transformer en images identiques des images qui étaient différentes.

6 Hautes et basses fréquences

(22)

Travaux Pratiques 11 Annuler la partie haute fréquence du spectre d'une image naturelle et commenter ce que devient l'image définie par ce nouveau spectre ?

Travaux Pratiques 12 A partir d'une image ayant un spectre particulier (choisir plutôt une image synthétique), changer la taille de cette image (allonger ou réduire la taille), cela peut se faire au moyen de imresize, que devient le spectre ?

7 Importance relative de la phase et du module dans le signal de transformée de Fourier

(8)

Travaux Pratiques 13 On s'intéresse plus souvent au module de la transformée de Fourier qu'à sa phase, pourtant celle-ci contient aussi une grande partie de l'information de l'image.

On considère ici une image naturelle image_n.

1. Calculer le module et l'argument de la transformée de Fourier de image_n puis d'une image formée d'un bruit blanc. Afficher l'image dont la transformée de Fourier a pour module celui du bruit blanc et pour phase celle d'image_n, cette image est notée image_nph. Afficher l'image dont la transformée de Fourier a pour phase celle du bruit blanc et pour module celui d'image_n, cette image est notée image_nmo. Commenter ces trois images.

8 Transformée en cosinus discrète

(9)

Travaux Pratiques 14 Calculez la transformée de Fourier puis en cosinus discrète de la même image image_s. Visualisez ces deux transformées. Mettez des échelles en cycles par image.

Travaux Pratiques 15 Il est possible de mettre en évidence les symétries et périodicités qui sont implicitement contenues dans le calcul d'une transformée de Fourier ou d'une transformée en cosinus discrète : il suffit d'intercaler des zéros entre chaque coefficient issu de la transformée, ainsi chacun de ces coefficients est remplacé par un groupe de quatre valeurs dont celle en haut à gauche prend la valeur du coefficient tandis que les autres sont nulles. Affichez les images ainsi obtenues et notées image_p1 et image_p2.

Cours 9 La transformée en cosinus discrète est définie par :

$$G_{k,l} = \alpha_k \alpha_l \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g_{m,n} \cos\left(\frac{\pi(2m+1)k}{2N}\right) \cos\left(\frac{\pi(2n+1)l}{2N}\right)$$

En ce qui concerne la transformée de Fourier discrète, les notions de symétries proviennent de ce que les deux expressions suivantes sont identiques à ceci près que l'implémentation fft2 de la première expression est calculée pour $(m,n) \in \{0..N-1\} \times \{0..N-1\}$ tandis l'implémentation fft2 de la deuxième expression est calculée pour $(m,n) \in \{0..2N-1\} \times \{0..2N-1\}$

$$g_{m,n} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} G_{k,l} e^{j2\pi \frac{km+ln}{N}}$$

$$g_{m,n} = \sum_{k=0, k}^{N-1} \sum_{l=0}^{N-1} G_{k,l} e^{j2\pi \frac{km+ln}{N}}$$

Cette assertion est aussi valable pour la transformée en cosinus discrète :

$$\begin{split} g_{m,n} &= \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \alpha_k \alpha_l G_{k,l} \cos \left(\frac{\pi (2m+1)k}{2N} \right) \cos \left(\frac{\pi (2n+1)l}{2N} \right) \\ g_{m,n} &= \sum_{k=0,\ k}^{2N-1} \underset{paire}{paire} \sum_{l=0\ l}^{2N-1} \underset{paire}{paire} \alpha_k \alpha_l G_{k,l} \cos \left(\frac{\pi (2m+1)k}{4N} \right) \cos \left(\frac{\pi (2n+1)l}{4N} \right) \\ \alpha_0 &= \frac{1}{\sqrt{N}} \ \textit{et} \ \alpha_k = \sqrt{\frac{2}{N}} \ \textit{si k est non-nul} \end{split}$$

<u>Exercices</u>

Cours 10 (11)

La transformée en cosinus discrète est définie pour une image carrée par :

$$G_{k,l} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \alpha_k \alpha_l g_{m,n} \cos\left(\frac{\pi(2m+1)k}{2N}\right) \cos\left(\frac{\pi(2n+1)l}{2N}\right)$$

avec

$$lpha_0=rac{1}{\sqrt{N}}$$
 et $lpha_k=\sqrt{rac{2}{N}}$ si k est non-nul

La transformée en cosinus discrète inverse est donnée par

$$g_{m,n} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \alpha_k \alpha_l G_{k,l} \cos\left(\frac{\pi(2m+1)k}{2N}\right) \cos\left(\frac{\pi(2n+1)l}{2N}\right)$$

On rappelle la relation trigonométrique $\cos(p)\cos(q) = \frac{1}{2}\cos(p+q) + \frac{1}{2}\cos(p-q)$.

Exercice 11 (11)

On cherche à vérifier que les formules de DCT et IDCT sont bien inverses l'une de l'autre. On observe que ces formules sont linéaires aussi il suffit de faire la démonstration pour des images ayant un seul pixel non-nul.

1. Calculez la transformée en cosinus discrète G_{kl} de $g_{nm} = 1$ si $n = n_0$ et $m = m_0$ et $g_{mn} = 0$ ailleurs.

- 2. On note h_{mn} la transformée en cosinus discrète inverse de G_{kl} . Exprimez h_{mn} comme le produit de deux sommes de cosinus. On note $a(m,m_0)=\sum_{k=0}^{N-1}\alpha_k^2\cos(\pi\frac{(2m_0+1)k}{2N})\cos(\pi\frac{(2m+1)k}{2N})$, montrez que $h_{mn}=a(m,m_0)a(n,n_0)$.
- 3. Montrez que $a(m, m_0)$ se met sous la forme d'une somme de cosinus :

$$a(m, m_0) = -\frac{1}{N} + \frac{1}{N} \sum_{k=0}^{N-1} \cos(\frac{\pi(m+m_0+1)k}{N}) \cos(\frac{\pi(m-m_0)k}{N})$$

- 4. En interprétant $\sum_{k=0}^{N-1} \cos(\frac{km}{N})$ comme la partie réelle d'une somme d'exponentielle, montrez que cette expression vaut N sauf quand m est multiple de 2N, 1 quand m est impaire et 0 quand m est paire.
- 5. Déduisez-en que $h_{mn} = 0$ sauf si $m = m_0$ et $n = n_0$ auquel cas $h_{mn} = 1$, c'est-à-dire que $h_{mn} = g_{mn}$.

Réponses fréquentielles des filtres

(23)

Cours 11 Le filtrage linéaire est une opération de convolution à 2 dimensions transformant une image en une autre en général de même taille. Il est défini par une matrice $[h_{m,n}]$ de taille $M \times N$ appelée masque de convolution (en général M=N). On cherche à calculer la réponse fréquentielle de ce filtre :

$$H(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h_{m,n} e^{-j2\pi(um+vn)}$$

H(u,v) est une fonction à valeurs complexes défini sur $\Re \times \Re$ et périodique de période 1 pour chacune de ses variables, u est appelée la fréquence spatiale horizontale et v la fréquence spatiale verticale. On la représente généralement sur $[-0.5, 0.5] \times [-0.5, 0.5]$. La fonction freqz2 permet d'obtenir cette réponse fréquentielle. Une autre solution pour calculer H(u,v) est de mettre le masque au centre d'une matrice nulle de taille par exemple 256 × 256 puis de calculer la transformée de Fourier discrète:

En effet, en posant M=N=256, les coefficients de cette transformée de Fourier discrète sont :

$$H(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} h_{m,n} e^{-j2\pi(\frac{km+ln}{N})}$$

On observe alors que chaque fréquence spatiale (u,v) peut être approchée par $(\frac{k}{N},\frac{l}{N})$. La convergence de l'algorithme provient alors de la continuité de la fonction exponentielle complexe : $|H(u,v)-H(k,l)| \leq \sum_{m,n} |h_{m,n}| 2\pi (m|u-\frac{k}{N}|+n|v-\frac{l}{N}|).$

$$|H(u,v) - H(k,l)| \le \sum_{m,n} |h_{m,n}| 2\pi (m|u - \frac{k}{N}| + n|v - \frac{l}{N}|)$$

Travaux Pratiques 16 On cherche maintenant à afficher le module de la réponse fréquentielle.

1. Afficher le module du spectre (|H(u,v)|) du filtre moyenneur 3×3 de masque $1/9 \star ones$ (3, 3) à l'aide du premier algorithme (et de façon facultative avec le second algorithme) en le représentant avec une image (en blanc les intensités élevées et en noir les intensités sombres).

Version Matlab du premier algorithme :

```
[H, Wx, Wy] = freqz2 (ones (3) /9);
figure (1); subplot (121); surf (Wx/2, Wy/2, abs(H));
```

Version Matlab du deuxième algorithme :

```
im=zeros(256);
im(127:129,127:129) = ones(3)/9;
figure(1); subplot(122);
[FreqX, FreqY] = meshgrid (-128:127, -128:127);
spm=abs(fftshift(fft2(im)));
surf(FreqX(1:5:end,1:5:end),FreqY(1:5:end,1:5:end),spm(1:5:end,1:5:end));
```

2. Pourquoi ne pourrions-nous pas utiliser ici l'instruction plot.

3. Afficher sous forme d'une image le module de la réponse fréquentielle du filtre

$$\left[\begin{array}{ccc}
2 & 1 & 0 \\
1 & 0 & -1 \\
0 & -1 & -2
\end{array}\right]$$

Exercices

Exercice 12 (12) On considère un filtre de masque de taille 2×2 définie par

$$[h_{m,n}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{11}$$

Calculez la réponse fréquentielle H(u,v). Représentez avec trois couleurs différentes le lieu des fréquences (u,v) pour lesquelles |H(u,v)| est maximum et minimum avec $u,v\in [-1,1]$ et le profil de $(u,v)\mapsto |H(u,v)|$ le long de u=v=f et $f\in [-1,1]$.

Cours 12 (15)

La transformée de Fourier d'une suite bidimensionnelle non-périodique est une fonction périodique de période égale à 1 suivant u et aussi suivant v, elle est définie par

$$\mathcal{T}F[g_{mn}](u,v) = \sum_{m,n} g_{mn} e^{-j2\pi(mu+nv)}$$
 (12)

On pourrait être tenté de normaliser par MN, mais dans ce cas il faudrait multiplier par MN la transformée inverse de Fourier. La transformée de Fourier inverse est définie par

$$\mathcal{T}F^{-1}[G(u,v)]_{mn} = \int_{u=-\frac{1}{2}}^{\frac{1}{2}} \int_{v=-\frac{1}{2}}^{\frac{1}{2}} G(u,v)e^{j2\pi(mu+nv)} dudv$$
 (13)

L'égalité de Parseval énonce que la puissance d'un signal déterministe peut se calculer à la fois à partir de g_{mn} et à partir de G(u,v).

$$\sum_{m} \sum_{n} g_{mn}^{2} = \int_{u=-\frac{1}{2}}^{\frac{1}{2}} \int_{v=-\frac{1}{2}}^{\frac{1}{2}} |G(u,v)|^{2} du dv$$
(14)

Les filtres de Sobel sont définis par

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 0 \\ 2 & 0 & -2 \\ 0 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & 1 \\ -2 & 0 & 2 \\ -1 & -2 & 0 \end{bmatrix}$$

On peut noter que les filtres de Sobel horizontaux et verticaux sont en fait séparables, c'est-à-dire que leur masques sont le produit de deux matrices plus simples. En fait il y a 4 autres filtres de Sobel qui sont les symétriques de ceux-ci.

Exercice 13 (Réponse fréquentielle du Sobel horizontal) (15)

- 1. Calculez la réponse fréquentielle du filtre de Sobel horizontal.
- 2. Tracez les profils horizontaux et verticaux du module de cette réponse fréquentielle avec une échelle en cycles par image (l'image est supposée être composée de 256 × 256 pixels).

Travaux pratiques

10 Analyse statistique du signal image

(10)

Cours 13 Le profil d'intensité d'une image en niveaux de gris correspond à une ligne souvent horizontale traversant l'image, il s'agit de la courbe des intensités de chaque pixel présent sur cette ligne. Par exemple le profil montré sur la figure 2 d'une image en niveaux de gris montrée sur la figure 1. Les pixels concernés par le profil sont montrés sur la figure 3. L'instruction axis permet de s'assurer de la maîtrise de l'axe des ordonnées (i.e. ne pas laisser qu'il y a de grandes fluctuations, alors qu'en réalité il n'y en a pas).

```
im=double(imread('coins.png'))/255;
figure(1); imshow(im);
figure(2); plot(1:size(im,1),im(:,50)); axis([-inf inf 0 1]);
im1=im; im1(:,50)=max(max(im));
figure(3); imshow(im1);
```

Travaux Pratiques 17 Choisir une image naturelle image_n ayant des contours précis.

- 1. Représentez un profil horizontal de l'image choisie.
- 2. On simule l'effet intégrateur des éléments photosensibles de la caméra en moyennant les intensités des pixels contenus dans une fenêtre représentant un élément CCD. Cette simulation se fait à l'aide d'une opération de convolution avec une fonction rectangle 2D (par exemple avec filter2 (masque, image)). Afficher cette image notée image bf. Pour le masque on peut utiliser ones (6)/36.
- 3. Le bruit d'acquisition est supposé ici additif. Pour le simuler on suppose que l'intensité de chaque pixel est modifiée par l'ajout d'un bruit blanc qui suit une loi gaussienne centrée (par exemple avec un écart-type de 1/30). Cela se fait sous Matlab avec randn qui permet de créer une image de même taille que l'image de départ et qui contient toute les variations. Afficher cette image est notée image_bg.
- 4. La fonction imnoise permet de simuler d'autres bruits. Simuler le bruit spot (Salt & Pepper). Afficher cette image, notée image_bs.
- 5. Tracez pour chacune des images obtenues les profils en s'assurant qu'à chaque fois les profils concernent les mêmes pixels et que l'axe des ordonnées a les mêmes graduations et comparez les profils obtenus avec le profil de l'image d'origine.

Les images ainsi obtenues image_bf, image_bg et image_bs seront utilisées dans le TP 18.

Exercices

Cours 14 (9)

Soit U une loi uniforme sur l'intervalle [a,b]. alors la densité spectrale, son espérance et sa variance sont données par :

$$f_U(x) = \frac{1}{b-a} \mathbf{1}_{[a,b]}(x) \quad E[U] = \frac{a+b}{2} \quad Var(U) = \frac{(b-a)^2}{12}$$
 (15)

La loi binômiale B(N,p) est la distribution de probabilité discrète du nombre de succès dans une suite de N expérimentations. Chacune de ces expérimentations ayant une probabilité p de succès. Si X suit une telle loi alors $P(X=k)=C_N^kp^k(1-p)^{N-k}$, E[X]=Np et Var(X)=Np(1-p).

En effet si X=k, cela signifie qu'il y a k expérimentations qui ont été un succès et N-k qui ont été un échec, aussi chaque expérimentation qui conduit à X=k a une probabilité de $p^k(1-p)^{N-k}$. Le nombre de façons de distribuer k succès parmi N expérimentations est donné par C_N^k . La deuxième et la troisième assertion sont facile à démontrer si on ne considère qu'une seule expérimentation : $E[X]=1\times p+0\times (1-p)$ et $Var(X)=E[X^2]-E[X]^2=1^2\times p+0^2\times (1-p)-p^2=p(1-p)$. En fait les différentes approximations sont indépendantes aussi l'espérance et la variance de chacune de ces expérimentations s'ajoutent.

La loi multinômiale $M(n, p_1 \dots p_m)$ est une généralisation de la loi binômiale. Cela consiste à procéder à n tirages indépendants qui sont à valeurs dans $\{1 \dots m\}$ avec les probabilités respectives $p_1 \dots p_m$. Les variables aléatoires M_i donnent le nombre de tirage qui ont donné la valeur $i \in \{1 \dots m\}$.

$$P(M_1 = x_1 \dots M_m = x_m) = \begin{cases} \frac{n! p_1^{x_1} \dots p_m^{x_m}}{x_1! \dots x_m!} & \text{si } \sum_i x_i = n \\ 0 & \text{sinon} \end{cases}$$

$$E[M_i] = np_i \quad Var[M_i] = np_i(1 - p_i) \quad Cov(M_i, M_j) = -np_i p_j \text{ si } i \neq j$$

où! désigne la factorielle.

La justification est la même que pour la loi binômiale. Les différents ensembles de tirages qui sélectionnent x_1 fois 1, x_2 fois 2 ... x_m fois m ont tous une probabilité $p_1^{x_1} \dots p_m^{x_m}$. Il y a $\frac{n!}{x_1! \dots x_m!}$ tirages de ce types parce que c'est le nombre de répartition de n éléments répartis dans m classes avec x_1 éléments dans la première, ... x_m éléments dans la dernière classe. Pour chaque tirage, on $E[M_i] = p_i$, $Var[M_i] = E[M_i^2] - E[M_i]^2 = 1^2 \times p_i + 0^2 \times (1-p_i) - p_i^2 = p_i(1-p_i)$, $Cov[M_i, M_j] = E[M_iM_j] - E[M_i]E[M_j] = 0 \times 1 - p_ip_j = -p_ip_j$. L'indépendance entre les tirages fait que les espérances, les variances et les covariances s'ajoutent.

Exercice 14 (9)

Le bruit de Salt & Pepper transforme chaque pixel en un pixel blanc ou noir avec une probabilité p pour chacun. On étudie les propriétés statistiques du bruit (Salt & Pepper).

- 1. Donnez les principales étapes d'un programme qui simule le bruit (Salt & Pepper).
- 2. On considère une image uniformément grise. Exprimez au moyen de deux variables aléatoires la moyenne de l'image dégradée.
- 3. Calculez l'espérance de l'erreur quadratique moyenne et de l'erreur absolue moyenne entre l'image grise d'origine et de l'image dégradée.

Travaux pratiques

11 Amélioration et restauration du signal image

(16)

Cours 15 Par opposition au filtrage non-linéaire, le filtrage linéaire ne fait pas apparaître de puissance sur une fréquence là où il n'y en avait pas. En revanche, il permet d'augmenter ou de diminuer la puissance sur telle fréquence. Les filtres linéaires utilisés ici consistent à remplacer la valeur de chaque pixel par une moyenne pondérée calculée avec les pixels voisins. En général, Ces filtres sont définis par une matrice, appelée masque, souvent de petite taille qui correspond aux coefficients de la moyenne pondérée. Les fonctions Matlab utiles sont filter2, fspecial, freqz2.

Les filtres considérés dans cette section permettent de lisser une image. Ce sont des passe-bas normalisés de sorte que la somme des coefficients du masque est égale à 1. Ces filtres peuvent aussi servir à restaurer une image bruitée, c'est-à-dire à se rapprocher de l'image d'origine au moyen d'opérations sur l'image bruitée.

Dans ce contexte, on définit une mesure de restauration.

Le taux d'amélioration de l'erreur quadratique moyenne (improved mean square normalized error en anglais):

$$TAEQM = \frac{\sum_{m} \sum_{n} (g_{m,n}^{o} - g_{m,n}^{r})^{2}}{\sum_{m} \sum_{n} (g_{m,n}^{o} - g_{m,n}^{b})^{2}}$$

où $[g_{m,n}^o]$, $[g_{m,n}^b]$, $[g_{m,n}^r]$ sont l'image origine, bruitée et restaurée.

TAEQM=std2(image_o-image_r)/std2(image_o-image_b);

Travaux Pratiques 18 On reconsidère les images image bf, image bg, image bs obtenues lors du TP 17.

- 1. Le masque du filtre moyenneur (ou average en anglais) est une matrice constante de somme égale à 1. Appliquer le filtre moyenneur de tailles 3×3 et 5×5 sur image_bs puis sur image_bg. Les images résultantes sont notées image_bsrm et image_bgrm. Commenter. Calculer le taux d'amélioration de l'erreur quadratique moyenne (TAEQM). Commentez sur le fait que tel TAEQM est plus grand ou plus petit que 1, pourquoi cette valeur 1 est particulière ?
- 2. Le filtre gaussien isotrope a des coefficients qui reproduisent approximativement une surface gaussienne d'écart-type σ. On obtient le masque d'un tel filtre avec la commande Matlab. Par exemple pour une taille 5x5 et un écart-type de 1,

```
h=fspecial('gaussian',[5 5],1);
```

Appliquer un tel filtre sur image_bs puis sur image_bg en choisissant convenablement σ (voisin de 1) et en choisissant la taille de masque de 3×3 ou 5×5 . Les images restaurées sont notées image_bsrg et image_bgrg. Commenter pour chaque image l'efficacité relative du filtre gaussien par rapport au filtre moyenneur. Calculer le taux d'amélioration de l'erreur quadratique moyenne (TAEQM) et commenter en comparant les TAEQM sur les mêmes images déformées mais restaurées avec des techniques différentes.

3. Représenter la réponse en fréquence du filtre moyenneur 3×3 et celle d'un filtre gaussien de même taille, leurs représentations graphiques sont notées image_tfm et image_tfg, respectivement. Commenter en rapport avec image_stf, image_bstf, image_bstf, image_bftf.

Exercices

Exercice 15 (31) *On se donne une image définie par :*

$$[g_{mn}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(16)$$

- 1. Calculez l'image obtenue en appliquant le filtre de masque [0.5 0.5].
- 2. Calculez l'image obtenue en appliquant le filtre défini par :

$$[h_{m,n}] = \frac{1}{2} \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]$$

3. Pourquoi d'une manière générale les masques de convolution utilisés pour lisser sont normalisés de telle sorte que $\sum_{m,n} h_{m,n} = 1$?

Travaux pratiques

12 Filtrage non-linéaire pour la réduction du bruit

(17)

Cours 16 Il existe des filtres non-linéaires pour diminuer un bruit spécifique : les filtres médians et médians à centre pondérés. Le filtre médian consiste à remplacer le niveau de gris de chaque pixel d'une image par la médiane d'un vecteur formé des niveaux de gris des pixels voisins. On pourra utiliser pour cela nlfilter et median sous Matlab.

```
A = imnoise(double(imread('cameraman.tif'))/255,'salt & pepper');
fun = @(x) median(x(:));
B = nlfilter(A,[3 3],fun);
figure(1); imshow(A),
figure(2), imshow(B),
```

Travaux Pratiques 19 Appliquer ce filtre sur l'image naturelle choisie déformée par un bruit Salt & Pepper (image_bs du TP 17), en choisissant une taille adéquate. L'image résultante est notée image_bsm. Décrire les déformations de l'image dûes au filtre et comparer avec image_bsrm et avec image_bsrg du TP 18.

Cours 17 Le filtre médian à centre pondéré consiste aussi à remplacer le niveau de gris de chaque pixel par la médiane d'un vecteur formé des niveaux de gris des pixels voisins et dans lequel l'élément central est répété plusieurs fois.

```
A = imnoise(double(imread('cameraman.tif'))/255,'salt & pepper');
fun = @(x) median([x(5)*ones(8,1); x(:)]);
B = nlfilter(A,[3 3],fun);
figure(1); imshow(A);
figure(2); imshow(B)
```

Travaux Pratiques 20 Appliquer ce filtre sur image_bs en indiquant la pondération choisie. L'image résultante est notée image_bsmp. Commenter.

Exercices

Exercice 16 (32) *On se donne une image définie par :*

$$[g_{mn}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(17)$$

1. Calculez l'image obtenue en appliquant le filtre médian de taille 2×2 .

13 Apparition de fausses couleurs lors de l'application d'un moyenneur

Travaux pratiques

(13)

Travaux Pratiques 21 Construisez une image synthétique avec deux couleurs de taille 64×64 . Appliquez à cette image une filtre moyenneur de taille 10×10 en l'appliquant successivement aux composantes rouges, vertes et bleues, puis reformer l'image à partir des trois composantes ainsi filtrées. Refaites l'expérience en utilisant différentes couleurs et pour chaque couple de couleurs, notez la couleur intermédiaire. Décrivez les résultats obtenus et commentez sur le caractère éventuellement inattendu des couleurs intermédiaires ?

14 Couleur, luminance, espace de couleur

(12)

Cours 18 Il existe différents espaces de couleurs dans lesquels cette transformation peut se faire, mais dans tous les espaces, il y a forcément deux signaux de chrominance qui comportent une information sur la teinte et la saturation (proportion de blanc) et un signal de luminance (sombre ou claire). Les fonctions rgb2ycbcr et ycbcr2rgb peuvent être utilisées. La transformation d'une couleur en un gris peut être approché par L=0.3*R+0.57*G+0.11*B. Géométriquement, il s'agit d'une projection d'un point de l'espace sur une droite, L indiquant la position sur cette droite.

- **Travaux Pratiques 22** 1. Choisissez une image naturelle en couleur. Calculer le signal luminance d'une image couleur, d'abord en extrayant les composantes rouge, puis verte puis bleue, et ensuite en utilisant seulement rgb2gray.
 - 2. Afficher successivement en niveau de gris les signaux de luminance et chrominance. Puis reformer l'image couleur et afficher l'image obtenue qui doit être identique à celle du départ.

15 Bruits additionnés

(28)

Travaux Pratiques 23 On considère une image couleur. On ajoute un bruit gaussien sur chacun des signaux de chrominance et on reforme l'image. Observer l'image obtenue. On ajoute un bruit gaussien sur le seul signal de luminance et on reforme l'image. Observer la nouvelle image obtenue. On ajoute un bruit gaussien sur chacune des composantes rouge, vert et bleu de l'image. Observer l'image obtenue en couleur puis en noir et blanc.

Exercices

Exercice 17 (25) On considère ici une approximation de l'espace YCbCr: Y=0.2125R+0.7154G+0.0721B, que l'on pourra noter $Y=c_{yr}R+c_{yg}G+c_{yb}B$. et les informations de chrominance par les grandeurs $C_R=R-Y$ et $C_B=B-Y$, appelées aussi différences de couleur.

- 1. On suppose que les valeurs de R,G, B évoluent entre 0 et 1. Entre quelles valeurs extrèmes évolue Y?
- 2. Exprimez C_R et C_B en fonction uniquement des canaux R, G et B. Quelle est la plus grande valeur positive prise par CR? Pour quel triplet (R,G,B) est-elle obtenue? Quelle est la plus petite valeur négative prise par CR? Pour quelle triplet (R,G,B) est-elle obtenue? Que peut-on déduire sur ce que représente l'axe portée par C_R ? Lorsque C_R évolue entre ses deux valeurs extrêmes, entre quelles couleurs se déplace t-on?

- 3. Mêmes questions pour C_B
- 4. Soit une couleur A de valeur R = G = B = 0.5. Quelles sont les valeurs de Y, C_R et C_B ? Quel est l'aspect de cette couleur?
- 5. On considère dans l'espace R, G et B, S stimuli colorés A(0.5,0,0) (rouge sombre), B(0.5,0.5,0) (jaune-vert sombre) et C(1,0,0) (rouge). Les distances AB et AC sont identiques. Ces deux distances correspondentelles à des distances de perception également semblables ?
- 6. On considère une image remplie de la couleur de fond A définie par R=0.5, G=0.5, B=0.5, et présentant un objet constitué de la couleur B de premier plan définie par Y=0.5, $C_R=0.5$, $C_B=0$, il s'agit d'un rose vif. Que donnerait l'acquisition en niveaux de gris de cette image?
- 7. On réalise une image carrée de 100 pixels de coté avec un dégradé vertical linéaire depuis la couleur B vers la couleur A. Représenter les histogrammes définis dans les plans Y, C_R , et C_B . Quelle serait l'allure des histogrammes des plans R, G, B?

Exercice 18 (26) On considère les espaces HSL et HSV définis par les transformations suivantes depuis un espace RGB de départ. Les modèles HSL et HSV ont été définis en 1978 et sont utilisés dans de nombreux logiciels de retouche et de traitement d'images. L'espace HSV est parfois aussi notée en France TSV. L'espace HSV est défini par les équations (18, 19, 20). L'espace HSL est défini par les équations (18, 21, 22).

La définition de la teinte (hue) est commune aux deux modèles :

$$H = \begin{cases} 0 & si \text{ max} = \min\\ (60^{\circ} \times \frac{G - B}{\max - \min} + 360^{\circ}) \text{ mod } 360^{\circ}, & si \text{ max} = R\\ 60^{\circ} \times \frac{B - R}{\max - \min} + 120^{\circ}, & si \text{ max} = G\\ 60^{\circ} \times \frac{R - G}{\max - \min} + 240^{\circ}, & si \text{ max} = B \end{cases}$$
(18)

où \max et \min désignent respectivement le maximum et le minimum de R, G et B.

Pour l'espace HSV, la définition de la saturation (saturation) et de la valeur (value) qui est une autre définition de la brillance.

$$S = \begin{cases} 0, & si \max = 0\\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & sinon \end{cases}$$
 (19)

$$V = \max (20)$$

Pour l'espace HSL, la définition de la saturation (saturation) et de la brillance (lightness) est donnée par :

$$S = \begin{cases} 0 & \text{si max} = \min \\ \frac{\max - \min}{\max + \min} = \frac{\max - \min}{2l}, & \text{si } l \le \frac{1}{2} \\ \frac{\max - \min}{2 - (\max + \min)} = \frac{\max - \min}{2 - 2l}, & \text{si } l > \frac{1}{2} \end{cases}$$

$$(21)$$

$$L = \frac{1}{2}(\max + \min) \tag{22}$$

- 1. Soit 3 couleurs A(100) (rouge), B(110) (jaune) et C(000) (noir) dans l'espace RGB. Donnez leurs coordonnées dans l'espace HSL puis dans l'espace HSV.
- 2. Quelles sont les règles permettant de dire quelle est la couleur dominante dans l'espace RGB à partir de la teinte d'une couleur ?
- 3. Retrouvez les composantes RGB de la couleur définie en HSL par $(30^{\circ}, 1, 1/2)$ et les composantes RGB de la couleur définie en HSV par $(30^{\circ}, 1, 1)$.

4. On note que pour les deux espaces de couleur HSL et HSV, les valeurs prises par S et L ou par S et V fixent de manière unique max et min des composantes colorimétriques de l'espace RGB. On considère maintenant un ensemble de couleurs définies en HSL par (H,1,1/2) avec H qui parcours [0°,360°], calculez les valeurs prises par les composantes colorimétriques dans l'espace RGB et dessinez une représentation en perspective de cette courbe, montrez que cette courbe se trouve sur les arêtes d'un cube.

Travaux pratiques

16 Filtrage linéaire pour réhausser les contours

(24)

Introduction 1 Les filtres considérés dans cette section, permettent de mettre en évidence les contours. Ces filtres transforment une image en une matrice de même taille dont les coefficients ne sont pas nécessairement positifs, il faut transformer ces coefficients (par exemple avec une transformation linéaire) avant de pouvoir afficher les résultats de ces filtres à l'aide d'une image en niveau de gris. Les filtres présentés ici sont définis par leur masque notés $[h_{m,n}]$. Ces filtres sont des passe-bande ou des passe-haut, c'est-à-dire qu'ils laissent les moyennes ou les hautes fréquences presqu'inchangées et qu'ils attenuent pratiquement les basses fréquences. En particulier la somme des coefficients du masque est nulle. Ces filtres se regroupent en deux types. Les filtres du premier ordre présentés sont directionnels et n'ont pas le même comportement suivant que les valeurs des pixels sont élevés d'un côté ou de l'autre du contour ou une zone discriminante. Les filtres du second ordre présentés ici ne sont pas directionnels, ils produisent une valeur élevée à chaque fois qu'il y a un contour ou une variation locale de niveau de gris.

Travaux Pratiques 24 On considère une image synthétique image_s, on pourra utiliser les fonctions fspecial, filter2, edge, freqz2 pour simuler ces filtres sous Matlab.

- 1. Appliquer un filtre gradient gradient à une image synthétique. En fait on applique souvent la valeur absolue, cela permet de traiter de la même façon une variation du blanc vers le noir et une variation du noir vers le blanc (polarité). La fonction Matlab gradient comporte en fait deux composantes. Quel est le masque associée à chacune des composantes ?
- 2. Appliquer 4 filtres de Sobel de façon à faire apparaître 4 parties distinctes des contours. Décrire les images résultantes. Les filtres de Sobel sont donnés page 24.
- 3. Le filtre de Marr et Hildreth est aussi appelé LOG pour Laplacien du[Of] Gaussien, on peut paramétrer σ l'écart-type de la fonction gaussienne. Appliquer à image_bg (bruit gaussien ajouté à une image synthétique) le filtre LOG en choisissant σ de façon à réduire l'effet nuisible dû au bruit. On note image_bglog, l'image résultante. Commenter le choix et l'avantage de ce filtre par rapport au précédent.
- 4. Visualiser le filtre LOG choisi, on note ce graphique image_logtf.

```
filtre_log=fspecial('log',5,2);
image_logtf=abs(freqz2(filtre_log));
figure(1); subplot(111); mesh(image_logtf);
```

Montrer qu'il est possible d'approximer le filtre LOG choisi en utilisant la différence de deux gaussiennes de paramètres différents. On note la visualisation de ce filtre image_gtf.

Appliquer ces deux filtres sur une image naturelle bruitée avec le même bruit gaussien, les images résultantes sont notées image_nlog et image_ng. Commenter les différences entre ces images.

5. Appliquer à image_bg le filtre laplacien. Le filtre laplacien est défini par

$$\frac{1}{8} \left[\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array} \right]$$

L'image résultante est notée image_bglap. Commenter les différences avec image_bglog.

Exercices

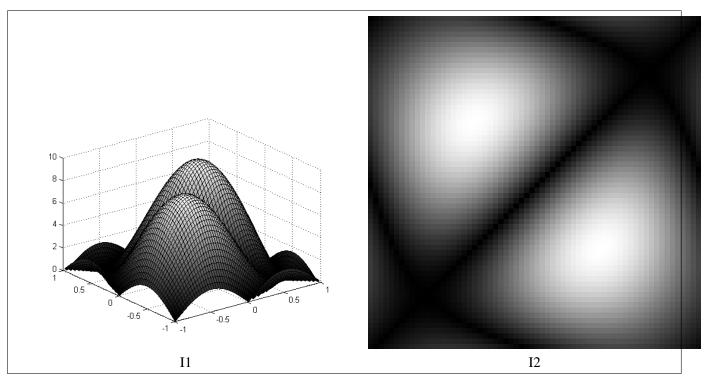


Figure 7: Spectres du filtre de Sobel présenté dans l'exercice 19

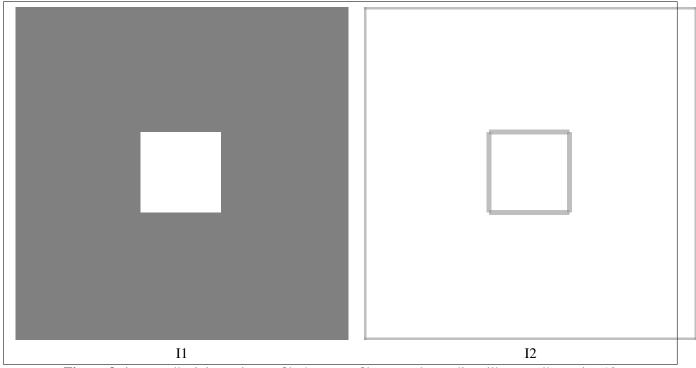


Figure 8: images d'origine et image filtrée par un filtre pseudo-gradient illustrant l'exercice 19

Exercice 19 (14) On se donne une image définie par :

$$g_{mn} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 1 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 1 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(23)$$

1. Appliquez le filtre de Sobel défini par

$$\begin{bmatrix}
 1 & 2 & 0 \\
 2 & 0 & -2 \\
 0 & -2 & -1
 \end{bmatrix}$$

Le module de sa réponse fréquentielle est montrée dans la figure 7.

- 2. Choisissez un seuil et calculez l'image de contour obtenue en seuillant la valeur absolue du résultat calculé précédemment.
- 3. Appliquez le filtre pseudo-gradient sur l'image $[g_{mn}]$. Calculez l'image résultante. Le résultat du filre pseudo-gradient est montré pour une image beaucoup plus grande dans la figure 8.
- 4. Choisissez un seuil et calculez l'image de contour obtenue en seuillant le résultat calculé précédemment.

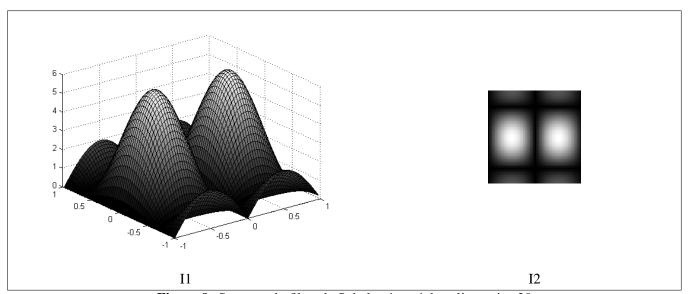


Figure 9: Spectres du filtre de Sobel présenté dans l'exercice 20

Exercice 20 (24) On considère le masque 3×3 de Prewitt parfois utilisé pour lisser le contour horizontal de l'image.

$$\nabla_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{24}$$

La transformée bidimensionnelle est définie par

$$H(z_1, z_2) = \sum_{k,l} h_{k,l} z_1^{-k} z_2^{-l}$$
(25)

1. Calculez la transformée en Z bidimentionnelle de la réponse impulsionnelle du filtre.

- 2. Pourra-t-on séparer le résultat en un produit de 2 filtres séparés H_1 et H_2 ?
- 3. Exprimez la réponse fréquentielle \hat{H}_1 et \hat{H}_2 . Dessinez à main levée les modules de ces réponses fréquentielles, s'agit-il de passe-bas, de passe-haut? Indiquez sur les graphiques les échelles en cycles par image si elles concernaient une image de taille 128×128 . Dessinez à main levée les fréquences pour lesquelles $\hat{H}(u,v)$ est nulle et les fréquences pour lesquelles $|\hat{H}(u,v)|$ est maximum. La figure 9 montre le module du spectre de ce filtre.
- 4. Ecrivez les masques de convolution de H_1 et H_2 . Retrouvez cette propriété de séparabilité.

Travaux pratiques

17 Filtres non-linéaires pour faire apparaître les contours

(25)

Travaux Pratiques 25 Il existe des filtres qui font apparaître les contours : pseudo-gradient à partir de deux filtres directionnels, pseudo-filtre passe-haut construit au moyen du filtre médian.

- 1. Appliquer le filtre pseudo-gradient à partir de deux réponses orthogonales du filtre Sobel sur l'image synthétique déformée par un bruit blanc gaussien image_bg. Comparer l'image résultante notée image_bgpg avec image_bglog et image_bglap.
- 2. La technique classique pour faire la synthèse d'un filtre linéaire passe-haut est de soustraire au filtre identité un filtre linéaire passe-bas adéquat. Appliquer cette technique pour faire la synthèse d'un pseudo-filtre passe-haut à partir du filtre médian considéré comme un passe-bas. Appliquer ce filtre aux images image_s et image_bs. Les images résultantes sont notées image_sph et image_bsph. Commenter et expliquer les résultats obtenus.

(35)

Introduction 2 Lors de cette séance, vous aurez chacun à choisir une image parmi un ensemble d'images qui vous seront proposées. Ces images sont globalement de bonnes qualité, cependant localement il y a des améliorations possibles, il peut s'agir d'enlever des caractères, des reflets, de changer l'éclairage dans un zone trop à l'ombre, de réduire un flou sur certaines régions, d'enlever un tramage. Il ne s'agit pas de trouver un traitement qui fonctionnerait pour toutes les images ayant ce type de problème, mais uniquement pour l'image choisie. A ce titre le fait qu'il s'agisse d'images colorées est une aide, puisque vous pouvez choisir l'espace de couleur et la composante lumineuse qui sont les plus adaptés au traitement souhaité. Ce sont des images de grandes tailles, c'est à vous de restreindre le traitement sur une portion de cette image. Un traitement est d'autant plus efficace qu'il s'applique à une portion précise de l'image, portion qui n'a aucune raison d'être un rectangle. C'est pour cela qu'une segmentation en régions vous sera utile.

Les programmes matlab disponibles sont :

```
growing1 : croissance par région avec critère sur l'écart-type de la région split4 : division par quadtree avec critère sur l'écart-type de la région split5 : division par quadtree avec critère abs(max-min) de la région imrag1 : graphe des liens d'adjascences merge3 : fusion avec graphe adjascence et critère sur l'écart-type de la région merge4 : fusion avec graphe adjascence et critère sur abs(max-min) de la région LabelMatch : à partir BW, met en blanc et noir les plus grandes régions imlabelclean1,2 : génère des régions connexes à partir de classes painting2 : colore chaque région avec la teinte correspondant à la moyenne des pixels
```

Deux images supplémentaires sont aussi disponibles : ispg.jpg et vaisselle.jpg Tous ces documents sont disponibles dans

```
http://www-12ti.univ-paris13.fr/~dauphin/PRG.7z
ou dans
http://www-12ti.univ-paris13.fr/~dauphin/PRG.exe
```

18 Un filtre qui préserve les contours

(39)

Cours 19 On étudie ici un filtre qui préserve les contours. Il ne s'agit pas d'un traitement linéaire. Ce filtre consiste à remplacer chaque pixel f_{mn} d'une image f par la moyenne des pixels de son voisinage compris entre $f_{mn} - \sigma$ et $f_{mn} + \sigma$, où σ est un des paramètres du traitement, et s'il n'y a aucun pixel entre $f_{mn} - \sigma$ et $f_{mn} + \sigma$ alors le pixel n'est pas modifié.

Voici quelques explications sur l'implémentation Matlab. La fonction nlfilter conssite à parcourir l'ensemble de l'image au moyen d'une fenêter, ici de taille 3×3 , et d'appliquer une fonction sur cette fenêtre. Par exemple on pourrait utiliser cette fonction nlfilter pour réaliser un filtrage par un moyenneur.

```
moyenneur=@(X) sum(X(:))/9;
im=nlfilter(im,[3 3],moyenneur);
On pourrait aussi assombrir l'image.
assombrir=@(X)0.9*X(2,2);
im=nlfilter(im,[3 3],assombrir);
```

On pourrait encore décaler l'image vers la droite.

```
decaler=@(X)X(2,3);
im=nlfilter(im,[3 3],decaler);
```

On pourrait aussi calculer le pseudo-gradient.

```
grad1=@(X) 0.5*(X(3,2)-X(1,2));
grad2=@(X) 0.5*(X(2,3)-X(2,1));
pseudoGradient=@(X) sqrt(grad1(X).^2+grad2(X).^2);
im=nlfilter(im,[3 3],pseudoGradient);
```

Voici un exemple d'instructions Matlab réalisant un nouveau traitement au moyen de la fonction nlfilter en utilisant la fonction newMean :

```
im=double(imread('coins.png'))/256;
im1=im+0.1*randn(size(im));
im2=filter2(ones(3)/9,im1,'same');
newMean=inline('sum((abs(X(:)-X(2,2))<0.1).*X(:))./sum(abs(X(:)-X(2,2))<0.1)','X');
im3=nlfilter(im1,[3 3],newMean);
figure(1); imshow([im im1; im2 im3]);
```

Travaux Pratiques 26 Appliquez cet exemple sur un exemple simple

Exercices

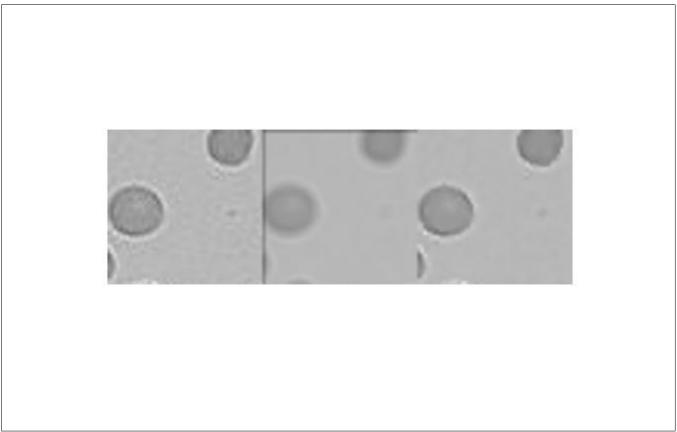


Figure 10: A gauche : image d'origine. Au milieu : traitement T1. A droite : traitement T2 avec $\sigma = 0.1$, exercice 21, (21)

Exercice 21 (21) On considère des voisinages carrés de tailles 3×3 . Soit un traitement T1 consistant a remplacer chaque pixel f_{mn} d'une image f par la moyenne des pixels de son voisinage. Soit un traitement T2 consistant a

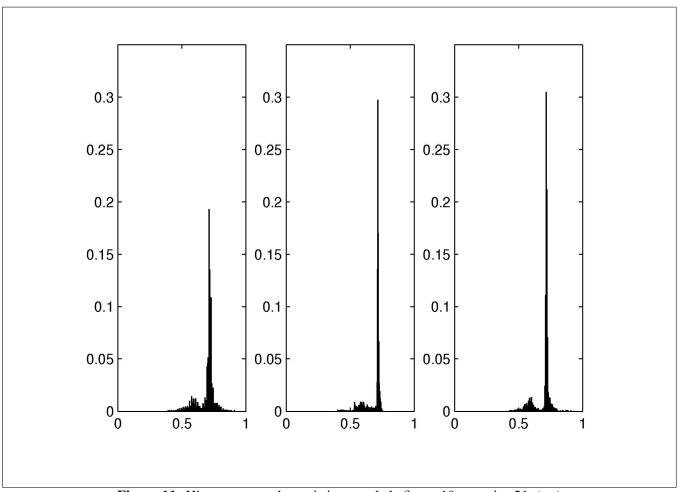


Figure 11: Histogrammes des trois images de la figure 10, exercice 21, (21)

remplacer chaque pixel f_{mn} d'une image f par la moyenne des pixels de son voisinage compris entre $f_{mn} - \sigma$ et $f_{mn} + \sigma$, où σ est un des paramètres du traitement, et s'il n'y a aucun pixel alors le pixel n'est pas modifié. La figure 10 (p. 36) montre à gauche une photo de globules rouges de sang, au milieu le résultat du traitement T1 et à droite le résultat de traitement T2 avec $\sigma = 0.1$. La figure 11 (p. 37) montre les histogrammes des trois images de la figure 10.

- 1. Montrez que le traitement T1 peut s'interprêter comme un filtrage.
- 2. Montrez que le traitement T2 n'est pas un traitement linéaire.
- 3. Comment expliquer l'impression de flou lié au traitement T1?
- 4. L'histogramme de gauche contient un pic principal, à quelle partie de l'image correspond-il?
- 5. L'histogramme de gauche a une bosse à gauche du grand pic, à quelle partie de l'image correspond-il ? L'histogramme a aussi une petite remontée à droite du pic, à quoi cela correspond-il ?
- 6. La distance entre les deux pics est de 0.15, comment cela peut-il nous aider à expliquer que les bords des cellules sont nets avec le traitement T2.
- 7. Commentez l'effet de T1 sur l'histogramme.
- 8. Commentez l'effet de T2 sur l'histogramme.

Un ingénieur remarque qu'on aurait pu obtenir sur cette image le même effet que le traitement T2 en faisant d'abord un seuillage puis en appliquant le traitement T1.

9. Expliquez la remarque de l'ingénieur et précisez les précautions qu'il faudrait prendre pour que cela fonctionne réellement aussi bien.

19 Seuillage

Travaux pratiques

(19)

Travaux Pratiques 27 Seuiller une image, c'est quantifier une image sur deux niveaux, ainsi les pixels dont le niveau de gris en dessous d'un seuil valent zéro tandis que les pixels dont le niveaux de gris est au dessus valent un. Le seuillage permet de segmenter en région (premier exemple qui suit) et de segmenter en contour (deuxième exemple qui suit).

1. A partir de l'image coins.png, fabriquer une image en noir et blanc tels que les points blancs correspondent aux sous, tandis que les points noirs représentent le fond. Montrer comment l'histogramme peut aider à choisir la valeur du seuil. Quelle est la valeur du seuil choisie? En ce qui concerne l'implémentation Matlab, on pourra utiliser ce type d'instruction.

```
figure (1); imshow(im.*(im>=0.3).*(im<0.4));
```

2. A partir de la même image coins.png, choisir un filtre qui réhausse les contours, appliquer ce filtre à cette image éventuellement avec un traitement adéquat, et enfin seuiller le résultat de l'application du filtre. Choisir un filtre et un seuil de façon que le résultat soit une image où ne subsiste que les contours des pièces de monnaie. Décrire la méthode utilisée et expliquer comment le seuil est choisi et quel compromis ce choix implique?

Exercices

Cours 20 (16)

Deux points sont connexes (au sens de la 4-connexité) l'un est le voisin du dessous, de gauche, de droite ou de dessus. Deux points sont connexes (au sens de la 8-connexité) si l'un est le voisin du desssous, de gauche, de droite ou de dessus, ainsi que d'en haut à gauche, en haut à droite, en bas à gauche et en bas à droite.

Exercice 22 (16) On se donne une deuxième image définie par

$$[g_{mn}] = \begin{bmatrix} 0 & 0 & 0.25 & 0.25 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0 & 0 & 0 & 0.25 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0 & 0 & 0.25 & 0.25 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0.5 & 0.5 & 0.5 & 0.7 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0.3 & 0.9 & 0.7 & 0.9 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.2 & 0.4 & 0.6 & 0.8 \\ 0.4 & 0.4 & 0.4 & 0.25 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0.5 & 0.5 & 0.4 & 0.6 & 0.6 & 0.4 & 0.5 & 0.6 \end{bmatrix}$$

$$(26)$$

1. Choisissez deux seuils de telle façon que l'image seuillée avec ces deux seuils fournissent trois régions connexes (4-connexité ici).

20 Segmentation en région

Travaux pratiques

(20)

Cours 21 La segmentation en région consiste à découper l'image suivant un critère en différentes régions connexes (c'est-à-dire composées de pixels qui sont voisins en un certain sens, le plus souvent dans le sens de la 4-connexité ou de la 8-connexité). Ainsi pour l'image coins.png, une segmentation en région possible consisterait à définir une région pour chaque pièce de monnaie ainsi qu'une région pour le fond de l'image.

Travaux Pratiques 28 Expliquez le fonctionnement du programme suivant :

```
im1=double(imread('coins.png'))/256;
im2=[0.5*ones(10,256); im1(:,1:256)];
qtim3=qtdecomp(im2,0.5);
im4=im2;
for dim=[2 4 8 16 32 64 128]
  [vals,r,c]=qtgetblk(im2,qtim3,dim);
  if (isempty(vals)==0)
    newvals=repmat(mean(mean(vals)),[dim dim]);
    im4=qtsetblk(im4,qtim3,dim,newvals);
  end;
end;
imshow(im4);
```

Commentez le résultat obtenu et expliquez les limites de cet algorithme.

21 Algorithme de segmentation en région par seuillage

Travaux pratiques

(41)

Cours 22 Un algorithme souvent utilisé pour choisir les seuils sur l'histogramme est kmeans. Cet algorithme réalise aussi le seuillage de l'image. Il faut aussi trouver à l'intérieur des classes les régions connexes : ceci peut se faire au moyen des outils morphologiques présents dans la fonction imlabelclean1.

Voici un exemple d'utilisation, la première phase réalise une partition des niveaux de gris.

```
im=double(imread('coins.png'))/256;
data=im(:);
%Initialisation
a=rand(1);
b=rand(1);
%Itération
for k=1:100
  indA=find(abs(data-a) < abs(data-b));</pre>
  indB=find(abs(data-a)>=abs(data-b));
  a=mean2(im(indA));
  b=mean2(im(indB));
end;
figure(1); imshow(im);
im2=0.5*ones(size(im));
im2(indA) = a;
im2(indB)=b;
figure(2); imshow(im2);
```

La deuxième phase consiste à subdiviser cette partition en régions connexes. Ici on ne conserve que les 13 régions les plus grandes.

```
im_s2=(im2>(a+b)/2);
im_s3=imlabelclean1(im_s2,13);
figure(3); imshow([im im_s2==1 im_s3/max(max(im_s3))]);
```

Pour faire rendre visible cette segmentation en région, chaque région est colorée avec la teinte moyenne des pixels présents dans la région.

```
im_s4=painting1(im,im_s3,zeros(size(im)));
figure(4); imshow(im_s4);
```

Souvent cette technique est utilisée non pas sur les niveaux de gris d'une image mais sur des descripteurs locaux de l'image comme la corrélation avec le pixel voisin.

Travaux Pratiques 29 Appliquez cette technique sur un exemple simple.

22 Lissage après une segmentation en région

Travaux pratiques

(40)

Cours 23 Lorsqu'on dispose d'une segmentation pertinente, l'amélioration de la qualité d'une image est plus facile. L'exemple qui suit montre deux traitements différents, un pour le fond de l'image (moyenneur de grande taille), l'autre pour les objets (moyenneur de courte taille et dont l'orientation est choisie pour minimiser la distortion). La séparation entre les deux se fait par seuillage d'après l'histogramme de l'image et après une petite concaténation avec des points voisins (fermeture morphologique).

```
im=double(imread('coins.png'))/256;
im1=im+0.1*randn(size(im));
figure(1); hist(im1(:),100);
seuil=0.4;
region=imopen(im1<seuil, ones(2));</pre>
ima=region.*im1;
ima21=filter2(ones(20),ima,'same');
ima22=filter2(ones(20),im1<0.5,'same');</pre>
ima2=zeros(size(im));
ima2(region==1) = ima21(region==1)./ima22(region==1);
imb=(1-region).*im1;
grad1=filter2([1;-1],imb,'same');
grad2=filter2([1 0; 0 -1], imb, 'same');
grad3=filter2([1 -1], imb, 'same');
grad4=filter2([0 1;-1 0],imb,'same');
orientation1=(1-region).*(abs(grad1)<min(min(abs(grad2),abs(grad3)),abs(grad4)));
orientation2=(1-region).*(abs(grad2)<min(min(abs(grad1),abs(grad3)),abs(grad4)));
orientation3=(1-region).*(abs(grad3)<min(min(abs(grad1),abs(grad2)),abs(grad4)));
orientation4=(1-region).*(1-orientation1-orientation2-orientation3);
imb21=filter2([1;1],imb,'same'); imb25=filter2([1;1],(1-region),'same');
imb22=filter2([1 0;0 1],imb,'same'); imb26=filter2([1 0;0 1],(1-region),'same');
imb23=filter2([1 1],imb,'same');imb27=filter2([1 1],(1-region),'same');
imb24=filter2([0 1;1 0],imb,'same');imb28=filter2([0 1;1 0],(1-region),'same');
imb2=zeros(size(im));
imb2 (orientation1==1) = imb21 (orientation1==1) ./imb25 (orientation1==1);
imb2 (orientation2==1) = imb22 (orientation2==1) ./imb26 (orientation2==1);
imb2 (orientation3==1) = imb23 (orientation3==1) ./imb27 (orientation3==1);
imb2 (orientation4==1) = imb24 (orientation4==1) ./imb28 (orientation4==1);
im2=ima2+imb2;
figure(2); imshow([im im1; region im2]);
```

Sur cet exemple, le traitement améliore un peu l'image bruitée, mais il est manifeste que le traitement pourrait être amélioré avec une meilleur segmentation.

Travaux Pratiques 30 Appliquez cette technique sur un exemple.

23 Segmentation en région par extraction des contours

Travaux pratiques

(26)

Cours 24 L'algorithme suivant présente une technique de segmentation en région qui consiste à tout d'abord extraire des contours pertinents de l'image, puis à trouver les régions qui sont délimitées par ces contours.

```
iml=double(imread('coins.png'))/256;
%iml=imresize(double(rgb2gray(imread('vaisselle.jpg')))/256,[128 128]);
%détection de contour,
contour=edge(im1,'log',0.0001,5);
%dilatation du contour
se=[0 1 0; 1 1 1; 0 1 0];
contour=imdilate(contour,se);
%inversion de l'image
```

```
im_region=1-contour;
%croissance de région connexe
im_label=im_region;
last=0; im_label_new=zeros(size(im_label));
for k=1:max(im_label(:))
   [B,L]=bwlabel(im_label==k);
   B=B+last.*(B>0);
   im_label_new=im_label_new+B; last=L+last;
end;
figure(1); imshow(im_label_new/max(im_label_new(:)));
```

Travaux Pratiques 31 Appliquez l'algorithme suivant sur une image, décrire le résultat, commentez. Expliquez le fonctionnement de l'algorithme suivant.

24 Segmentation en région par division successive de l'image puis fusion successive (*split & merge*)

Travaux pratiques

(37)

Cours 25 La segmentation par quadtree (encore appelée split) ne produit pas des régions ayant des contours lisses. Par contre on peut y parvenir en fusionnant les différentes régions obtenues. Cette fusion se fait ici en deux étapes, d'abord la création d'une liste de couples de régions susceptibles d'être fusionnées (en pratique il s'agit d'un arbre binaire), puis un éventuel fusion couple de régions par couple de régions lorsqu'un critère est vérifié.

La simulation est ici illustrée sur une image synthétique formée d'un disque avec un teinte très légèrement variable à l'intérieur et un teinte unie à l'extérieur.

En fait de manière très voisine, on aurait pu aussi illustrer la méthode sur une image synthétique formée d'un disque avec un bruit blanc et autour un bruit blanc coloré.

L'efficacité d'un algorithme de segmentation est d'autant plus grande que le critère qu'il utilise est adapté à l'image. Les algorithmes split & merge autorisent des critères portant sur l'ensemble de la région, comme un seuil sur la variance.

```
saturate=inline('max(0,min(1,x))','x');
m=meshgrid(-1:1/63.5:1); disc=(m.^2+(m.').^2<=0.5);
imd=disc.*m+(1-disc)*10;
Mx=max(max(imd(imd<10))); Mn=min(min(imd(imd<10)));
im=0.4+0.2*(imd-Mn)/(Mx-Mn).*(imd<10)-0.04*(1-disc);
imd=disc.*m+(1-disc)*10;
Mx=max(max(imd(imd<10))); Mn=min(min(imd(imd<10)));
im=0.4+0.2*(imd-Mn)/(Mx-Mn).*(imd<10)-0.04*(1-disc);
im_s1=split4(im, 0.01);
adj=imrag1(im_s1);
im_s2=merge5(im_s1,im,0.01,adj);
im_s3=LabelMatch(im_s2,disc);
%Affichage
figure(1);
subplot(121); imshow(im);
subplot(122); imshow(im_s3);
```

Travaux Pratiques 32 Appliquez cette technique sur un exemple simple.

Travaux pratiques

(38)

Cours 26 La segmentation par croissance de région consiste à choisir un pixel dans l'image, appelée graine (seed) puis à agglomérer les pixels voisins tant qu'un certain critère est satisfait. Lorsqu'il n'est plus possible de faire croître la région, l'algorithme choisit un autre point ailleurs dans l'image et le même processus recommence.

Dans la pratique les résultats obtenus par cet algorithme dépendent bien sûr du critère choisi, ils dépendent aussi de l'implémentation : comment choisit-on le point de départ, dans quel ordre cherche-t-on à agglomérer les pixels voisins.

```
im2=imresize(double((imread('ispg.jpg')))/256,[128 128]);
%im1=imresize(double(rgb2gray(imread('vaisselle.jpg')))/256,[128 128]);
im_s21=growing1(im2,0.08);
%En ne conservant que les 13 régions les plus grandes
im_s23=imlabelclean1(im_s21,13);
%Coloration
im_s24=painting2(im2,im_s23,zeros(size(im2)));
```

Travaux Pratiques 33 Appliquez cet exemple sur un exemple simple.

26 Restauration d'images par filtre de Wiener

Travaux pratiques

(36)

Cours 27 Dans le cadre de cette section, on cherche à restaurer une image floue éventuellement bruitée avec un bruit additif gaussien, mais avec cette particularité que l'on connaît exactement le flou et aussi le bruit.

```
im=double(imread('coins.png'))/256;
im=im(1:128,1:128);
[x,y]=meshgrid((0:8)/8,(0:8)/8);
mask=((x-0.5).^2+(y-0.5).^2<=0.5^2);
mask=mask/sum(sum(mask));
im1=filter2(mask,im,'same');
figure(1); imshow([im im1]);
```

Une première technique de restauration d'image consiste à appliquer le filtre inverse, dont la transformée de Fourier est égale à l'inverse de la transformée de Fourier du filtre flou. Dans la pratique, on calcule la transformée de Fourier du masque en plaçant celui-ci au centre d'une matrice nulle. On met un seuil et le filtre inverse se calcule de la façon suivante :

```
filtre_tf_inv=(filtre_tf>0.05).*(1./filtre_tf);

ImMask=zeros(size(im1));

ImMask(size(im1,1)/2-4:size(im1,1)/2+4,size(im1,2)/2-4:size(im1,2)/2+4)=mask;

filtre_tf=fft2(ImMask);

seuil=0.2;

[ind1,ind2]=find(filtre_tf>seuil);

filtre_tf_inv=1/seuil*ones(size(filtre_tf));

filtre_tf_inv(ind1,ind2)=1./filtre_tf(ind1,ind2);

im2=real(fftshift(ifft2(filtre_tf_inv.*fft2(im1))));

figure(1); imshow([im im1 im2]);
```

Le calcul précédent se justifie de la façon suivante.

On désigne par f, g, h, \hat{f} les quantités suivantes : l'image originale image_o, l'image dégradée image_of, le masque filtre, l'image restaurée notée image_or. Ainsi g est l'image déformée par le filtre h, tandis que \hat{h} est le filtre que l'on recherche pour restaurer l'image. Le modèle déterministe est donné par :

$$\begin{cases} g = h \star f & (d\acute{e}gradation \, sans \, bruit) \\ \hat{f} = \hat{h} \star g & (restauration) \end{cases}$$
 (27)

Le filtre \hat{h} que l'on recherche est celui qui minimise l'erreur quadratique entre l'image estimée notée \hat{f} et la vraie image notée f. Cette technique s'appelle LMS en anglais pour Linear Mean Square Error (voir cours de TSS).

$$EQ = \sum_{m,n} (\hat{f} - f)^2$$

La minimisation de l'erreur quadratique conduit au filtre inverse $\hat{H} = \frac{1}{H}$ (lorsque le filtre est inversible).

Pour améliorer la restauration, on suppose qu'en fait il a été rajouté un bruit blanc gaussien additif en plus du flou.

Pour tenir compte correctement de l'ajout du bruit blanc, il est nécessaire de rajouter dans le modèle ce qu'on sait de l'image originale. Cette information s'appelle un a priori. Ici il s'agit de la moyenne de l'image.

On désigne par f, b, g, h, \hat{f} , \hat{h} , les quantités suivantes : l'image originale image_o, le bruit blanc additif image_ofb-image_of, l'image dégradée image_ofb, le masque filtre, l'image restaurée image_or, la réponse impulsionnelle du filtre que l'on souhaite appliquer pour restaurer l'image.

L'information que l'on se donne sur f est sa moyenne sur l'image, notée μ_f et sa puissance P_f . Pour introduire cette information dans un modèle, on dit que $\xi = f - \mu_f$ est une réalisation d'un processus aléatoire gaussien blanc centré d'écart-type σ_f et indépendant du bruit b. Le modèle est :

$$\begin{cases} f = \mu_f + \xi & (a \text{ priori}) \\ g = h \star f + b & (observation) \\ \hat{f} = \hat{h} \star (g - \mu_f) + \mu_f & (restauration) \end{cases}$$
 (28)

Ainsi g est aussi une réalisation d'un processus aléatoire gaussien centré blanc d'écart-type σ_b . On cherche maintenant le filtre \hat{h} qui minimise l'erreur moyenne quadratique entre l'image estimée \hat{f} et la vraie image f.

Le filtre obtenu a pour transformée de Fourier

$$\hat{H} = \frac{\bar{H}}{|H|^2 + \frac{\sigma_b^2}{\sigma_x^2}}$$

```
moy=mean2(im1);
ImMask=zeros(size(im1));
ImMask(size(im1,1)/2-4:size(im1,1)/2+4,size(im1,2)/2-4:size(im1,2)/2+4)=mask;
filtre_tf=fft2(ImMask);
alpha=0.02;
filtre_tf_inv=conj(filtre_tf)./(alpha+abs(filtre_tf).^2);
im2=moy+real(fftshift(ifft2(filtre_tf_inv.*fft2(im1-moy))));
figure(1); imshow([im im1 im2]);
```

Le logiciel Matlab propose d'utiliser un préfiltrage (c'est-à-dire un filtrage de image_of avant d'appliquer le filtre de Wiener). Le préfiltrage proposé est donné par la fonction edgetaper et compense les effets de bords assez importants provoqués par l'utilisation de la transformée de Fourier discrète (fft2).

```
im1bis=edgetaper(im1, fspecial('gaussian', 50, 8));
moy=mean2(im1bis);
ImMask=zeros(size(im1));
ImMask(size(im1,1)/2-4:size(im1,1)/2+4, size(im1,2)/2-4:size(im1,2)/2+4)=mask;
filtre_tf=fft2(ImMask);
```

```
alpha=0.001;
filtre_tf_inv=conj(filtre_tf)./(alpha+abs(filtre_tf).^2);
im2=moy+real(fftshift(ifft2(filtre_tf_inv.*fft2(im1bis-moy))));
figure(1); imshow([im im1 im2]);
```

Travaux Pratiques 34 Appliquez ces techniques sur un exemple simple.

27 Filtres adaptatifs pour images non-stationnaires

Cours 28 Lorsqu'on applique un filtre linéaire invariant, c'est-à-dire un filtre dont les coefficients du masque sont identiques quel que soit l'endroit de l'image où on l'applique, cela revient à modéliser l'image comme un procesus stationnaire. En réalité, il serait plus juste de considérer l'image comme un processus non-stationnaire, une image source représente souvent des objets différents placés dans une scène.

Le raisonnement est modifié dans la mesure où l'on connaît localement f en général sa moyenne locale et sa puissance locale. C'est cela qui consiste le nouvel a priori.

Dans le modèle suivant, on analyse l'image au moyen d'une fenêtre glissante $W_{k,l}$ centré en un pixel (k,l), et on traite chaque élément (m,n) de $W_{k,l}$, parcourt chacun des pixels de l'image (k,l) et pour chacun, on se donne une fenêtre dont (k,l) est le centre et dont (m,n) sont les indices courants.

$$\begin{cases}
f(m,n) = \mu_f(k,l) + \xi(m,n) & (a priori) \\
g = h \star f + b & (observation) \\
\hat{f}(k,l) = \left(h'_{k,l} \star (g - \mu_f(k,l))\right)(k,l) + \mu_f(k,l) & (restauration)
\end{cases} \tag{29}$$

où $\mu_f(k,l)$ est une constante et vaut la moyenne de f sur la fenêtre ; $\hat{h}_{k,l}$ est la masque à appliquer à la fenêtre centrée en (k, l) pour calculer l'image restaurée en (k, l).

1. Le précédent raisonnement donne aussi de bons résultats lorsqu'on cherche à retirer du bruit sur une image

qui n'a pas été filtrée. Dans ce cas
$$H=1$$
 et la restauration consiste en $\hat{f}(k,l)=\mu_f(k,l)+rac{1}{1+rac{\sigma_b^2}{\sigma_f(k,l)^2}}(f-\mu_f(k,l))$. Le premier terme est un passe-bas tandis que le second

terme est un passe-haut multiplié par une fonction variable spatialement. Ceci s'appelle un filtre de Lee, il est implémenté sous Matlab sous le nom de wiener2. Ajouter un bruit blanc gaussien centré d'écart-type $\sigma_b = 0.2$ à image o pour former image ob. Appliquer ce filtre sans utiliser la fonction wiener 2. Le résultat est noté image_ol. Afficher aussi image_pd, l'image formée des composantes $\frac{1}{1+\frac{\sigma_b^2}{\sigma_f(k,l)^2}}$. Pour

estimer $\sigma_f(k,l)^2$ on pourra utiliser plutôt $<(g-< g>)^2>$ sans retrancher σ_b^2 .

```
im=double(imread('coins.png'))/256;
im=im(1:128,1:128);
im1=im+randn(size(im))*0.2;
Win=[5 5]; Mask=ones(Win)/sum(sum(Win));
ImMoy=filter2(Mask,im1,'same');
beta=1./(1+0.2^1./filter2(Mask,(im1-ImMoy).^2,'same'));
im2=ImMoy+beta.*(im1-ImMoy);
figure(1); imshow([im im1 im2]);
```

On observe que les bords sont bien préservés.

2. On applique maintenant le précédent raisonnement au cas où l'image bruitée a été filtré par un filtre connu. L'inversion du filtre se fait alors en calculant une transformée de Fourier locale et non plus sur l'ensemble de l'image. Voici les instructions Matlab permettant d'obtenir cela. S'il n'y a pas de bruit additif, le résultat est moins bon, ce qui n'est pas surprenant puisque l'information sur le filtre est moins prise en compte. En revanche cette méthode fonctionne dans une certaine mesure en présence de bruit.

```
enlarge=Q(X) [zeros(1, size(X, 2) +2); ...
  [zeros(size(X,1),1) X zeros(size(X,1),1)]; zeros(1,size(X,2)+2)];
im=double(imread('coins.png'))/256;
im=im(1:128,1:128);
[x,y] = meshgrid((0:8)/8, (0:8)/8);
mask=((x-0.5).^2+(y-0.5).^2<=0.5^2);
mask=mask/sum(sum(mask));
im1=filter2(mask,im,'same')+0.01*randn(size(im));
im1Moy=filter2(ones(11)/121, im1, 'same');
spectrum=fft2(enlarge(mask));
alpha=0.01;
processing=@(im, spectrum, alpha)[zeros(1,5) 1 zeros(1,5)]*...
  fftshift(ifft2(fft2(edgetaper(im, ones(3)/9)).*...
  conj(spectrum)./(abs(spectrum).^2+alpha^2/std2(im)^2)))*...
  [zeros(5,1); 1; zeros(5,1)];
im2=im1Moy+nlfilter(im1-im1Moy,[11 11],processing,spectrum,alpha);
figure(1); imshow([im im1 im2]);
```

Travaux Pratiques 35 Appliquez cette technique sur un exemple simple.

Exercices

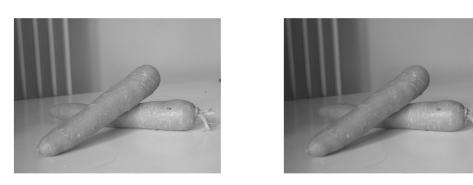


Figure 12: A gauche : la première photo. A droite : moyenne de 14 photos. Exercice 23, (20)

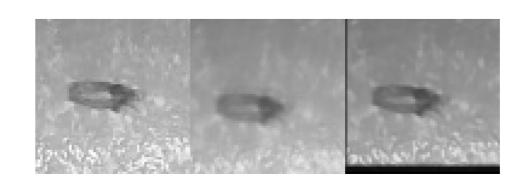


Figure 13: A gauche : détail de la première photo. Au milieu : moyenne des détails. A droite : moyenne après recalage par translation. Exercice 23, (20)

(exercice 23)

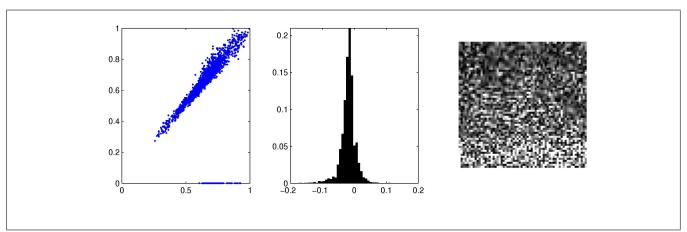


Figure 14: A gauche : niveau de gris des pixels de la photo 4 en fonction des niveaux de gris des pixels de la photo 3. Au milieu : histogramme des différences entre ces niveaux de gris. A droite : valeur absolue des différences entre les niveaux de gris multiplié par 20. Exercice 23,(20)

Tableau 1	
0 -1 -7 -11 -17 -20 -16 -21 -20 -20 -20 -20 -	20 -9
1 0 -4 -9 -16 -2 0 -2 0 0 -2 -2	0 -2
7 4 1 -4 -12 -12 -12 -16 -12 -13 -16 -16 -	16 -20
11 9 4 1 -8 -8 -8 -12 -8 -8 -9 -11 -	11 -16
17	-3 -8
20 2 12 8 0 0 -1 -4 0 0 -3 -3	-4 -8
16 0 12 8 0 1 0 0 0 0 0	-4 -5
21 2 16 12 4 4 0 0 0 0 0	0 -4
20 0 12 8 1 0 0 0 0 0 0	0 -4
20 0 13 8 1 0 0 0 0 0 -1 0	-2 -4
20 2 16 9 4 3 0 0 0 1 0 0	0 - 4
20 2 16 11 4 3 0 0 0 0 0	0 -4
20 0 16 11 3 4 4 0 0 2 0 0	0 -3
9 2 20 16 8 8 5 4 4 4 4	3 0
(exercice 23)	
Tableau 2	
1 0 8 8 8 8 1 8 1 1 4 1	1 0
0 1 9 8 9 0 1 0 -1 -1 0 0	1 0
-8 -9 0 0 0 1 0 4 4 4 4	3 0
-8 -8 0 0 0 0 1 0 3 5 8 4	4 0
-8 -9 0 0 1 2 2 0 4 4 8 5	4 0
-8 0 0 0 -2 1 0 0 4 4 4 4	3 0
-1 -1 -1 -1 -2 0 1 1 2 4 4 4	0 0
-8 0 0 0 0 0 -1 1 2 4 4 4	4 0
-1 1 -4 -3 -4 -4 -2 -2 1 -1 2 2	1 -4
-1 1 -4 -5 -4 -4 -4 1 1 0 1	0 -4
-4 0 -4 -8 -8 -4 -4 -4 -2 0 1 1	-2 -5
-1 0 -4 -4 -5 -4 -4 -4 -2 -1 -1 1	1 -4
-1 -1 -3 -4 -4 -3 0 -4 -1 0 2 -1	1 -4
0 0 0 0 0 0 0 4 4 5 4	4 1

Exercice 23 (20) La photo de gauche sur la figure 12 (p. 46) a été prise avec un appareil photo. L'ingénieur A propose d'améliorer la qualité de la photo en prenant plusieurs fois la même photo puis à moyenner les différentes images pour en récupérer une qui soit de meilleur qualité. Il affirme que dans la mesure où l'appareil photo n'a pas bougé, les différences entre les photos peuvent être modélisées par un unique bruit blanc gaussien additif dont la variance est réduite lorsqu'on fait la moyenne.

1. Expliquez pourquoi si on suit la modélisation de l'ingénieur A, la moyenne des photos est aussi perturbée par un bruit blanc gaussien additif dont la variance est 14 fois plus petite ?

La photo de droite de la figure 12 (p. 46) montre le résultat du procédé. En réalité cette image est plus floue et semble avoir été prise avec une luminosité différente. L'ingénieur B propose de se concentrer sur un détail de l'image (montré à gauche de la figure 13 (p. 46)) pour diminuer le coût de calcul, il propose aussi de réaliser un suréchantillonnage et de supprimer les photos trop différentes, c'est le cas des deux premières. La photo au milieu de la figure 13 est obtenue en moyennant sur les 12 autres photos. L'ingénieur C pense que le flou de la photo du milieu provient de ce que l'appareil photo a en réalité bougé très légèrement lorsqu'on a appuyé sur le boutton. Il propose, pour trouver les coordonnées du vecteur de translation à appliquer, de calculer le maximum de l'intercorrélation entre les images. Il précise qu'il n'est pas nécessaire de calculer ces intercorrélations au-delà des 10 premiers indices.

- 2. On considère une image g et cette même image mais translatée h, montrez que la somme des carrés des différences est minimale si l'intercorrélation est maximale.
- 3. Donnez la formule de l'intercorrélation. Comment faire pour que la mesure de l'intercorrélation ne pénalise pas le déplacement de l'image ?
- 4. Pourquoi n'est-il pas nécessaire de calculer au-delà des 10 premiers indices ?
- 5. Ecrivez la nouvelle formule de moyenne à appliquer.
- 6. Pourquoi un recalage peut-il être plus précis après un suréchantillonnage?

Les tableaux 1 et 2 indiquent les coordonnées des vecteurs de translation pour faire coïncider les images. Le tableau 1 correspond à la première coordonnée, tandis que le tableau 2 correspond à la deuxième coordonnée. Les deux composantes t_{cl} et s_{cl} située sur la colonne c et la ligne l des deux tableaux donnent le vecteur de translation (t_{cl}, s_{cl}) qui appliquée à l'image l permettent de la caler sur l'image c. A droite de la figure 13 se trouve l'image obtenue en moyennant les différents images de départs après recalage.

7. Pourquoi les tableaux 1 et 2 sont-ils presque anti-symétriques? Que nous informe les termes non-nuls sur les deux diagonales?

L'ingénieur A, déçu des résultats, cherche à savoir si la modélisation est juste. Pour chaque couple d'images de détail, il porte sur un graphique l'ensemble des pixels sous la forme de points dont l'abscisse est le niveau de gris du pixel sur la première image et l'ordonnée est le niveau de gris du pixel de la deuxième image (à gauche de la figure 14, p. 47). Il représente ensuite l'histogramme des différences entre les niveaux de gris sur une deuxième graphique (au milieu de la figure 14). Enfin il représente la différence entre les images (à droite de la figure 14). L'ingénieur A en conclu que pour certains couples d'images les conditions sont maintenant bien réalisées. L'ingénieur B affirme que ces expériences ne suffisent pas pour affirmer que la modélisation est correcte, parce que l'on observe que les variations entre les images et non les variations entre les images et la vraie image. Dans la réalité le détail au centre de l'image est n'est pas une forme convexe mais une entaille dont l'extérieur a noirci et présente la forme d'un arceau presque fermé. C'est la compression jpeg faite par l'appareil photo qui a déformé l'image. En fait toutes les images ont été obtenues avec le même appareil photo et les mêmes réglages et présentent le même biais. Plus généralement le fait d'augmenter la résolution d'une photo en utilisant les informations contenues dans d'autres photos du même sujet est un axe de recherche actuel appelé super-résolution.

- 8. Pourquoi nous montre-t-il que la distortion entre les deux images est bien modélisée par un bruit bruit additif ?
- 9. Que nous informe le graphique au milieu de la figure 14?
- 10. Que nous informe le graphique à droite de la figure 14?

(1)

Cours 29 Un histogramme d'une image numérique est une courbe qui représente le nombre de pixels (ou la proportion de pixels) dont le niveau de gris se situe entre telle et telle valeur. En pratique, pour faire un histograme on effectue une quantification des niveaux de gris sur N niveaux, puis on trace une courbe du nombre de pixels en fonction du niveau. Par extension, on peut définir un histogramme pour une image définie par une fonction de $\mathbb{R} \times \mathbb{R}$ dans \mathbb{R} , en quantifiant cette image sur N niveaux et en mesurant la portion de surface de l'image correspondant à chacun de ces niveaux. Si une image est une réalisation d'un bruit blanc suivant une certaine loi alors l'histogramme est souvent une bonne approximation de la densité de probabilité de cette loi.

Exercice 24 On considère une image carrée de taille $A \times A$ pixels, quantifiée sur 256 niveaux à valeurs dans [0,1]. Cette image contient en son centre un carré de taille $B \times B$ dont les pixels valent tous g_b . Les pixels du reste de l'image valent g_a . Pour simplifier on considère que A et B sont impaires.

- 1. Définissez l'image en utilisant l'opérateur $\mathbf{1}_{\mathbb{A}}$ (où $\mathbf{1}_{\mathbb{A}}(x) = 1$ si $x \in \mathbb{A}$ et $\mathbf{1}_{\mathbb{A}}(x) = 0$ sinon, x étant un élément d'un ensemble plus vaste contenant \mathbb{A}).
- 2. Ecrivez et tracez l'histogramme correspondant à cette image.

On ajoute un bruit additif gaussien blanc à cette image. La densité de probabilité f associée à la modification d'un pixel g'=g+b est $f(b)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\frac{b^2}{\sigma^2}}$. La modélisation repose sur une variable aléatoire continue que l'on applique ici à une image quantifiée.

3. Quelle est la probabilité qu'un unique pixel soit augmenté de 1/255.

Pour répondre à cette question et pour passer d'une description discrète à une description continue on pourrait donner trois significations à cette phrase :

- H1: Le pixel est passé de $\frac{m}{255}$ à $\frac{m}{255} + \frac{1}{255}$.
- H2 : Le pixel est passé de $\frac{m}{255}$ à un niveau de gris contenu dans $]\frac{m}{255} + \frac{1}{2\times255}, \frac{m}{255} + \frac{3}{2\times255}]$.
- H3: Le pixel avait un niveau de gris quelque part dans $\left[\frac{m}{255} \frac{1}{2 \times 255}, \frac{m}{255} + \frac{1}{2 \times 255}\right]$ il a maintenant un niveau de gris contenu dans $\left[\frac{m}{255} + \frac{1}{2 \times 255}, \frac{m}{255} + \frac{3}{2 \times 255}\right]$. On modélise l'ignorance du niveau de gris par une loi uniforme.

En fait la première hypothèse conduit à une probabilité nulle qui est un résultat absurde on ne peut donc utiliser cette hypothèse. La deuxième et la troisième conduisent à des résultats voisins, on utilise donc la deuxième hypothèse plus simple.

- 4. Quelle est la probabilité que l'ensemble de l'image devienne une image uniforme?
- 5. Quelle est l'espérance du nombre de pixels dont le niveau de gris a décru de 1/255?

Dans cette question, tout se passe comme si on faisait un tirage de Bernouilli où on se donne une certaine probabilité que certains pixels subissent une transformation, on sait alors que l'espérance du nombre de pixels subissant cette transformation est donnée par le nombre de pixels de l'image multiplié par cette probabilité.

6. A partir des calculs précédents, calculez un histogramme moyen de l'image modifiée par le bruit additif.

On définit l'histogramme par $H = -\sum_{i=0}^{255} p_i \log_2(p_i)$ où p_i sont les probabilités qu'un pixel ait le niveau de gris $g_i = \frac{i}{255}$.

7. Calculez l'entropie de l'image sans bruit additif.

8. Calculez l'entropie de l'image lorsqu'on rajoute un bruit impulsionnel, ici pour chaque pixel il y a une probabilité q = 0.05 que celui-ci vaille 1.

Pour une image de taille $M \times N$, on définit ici la matrice de coocurrence par

$$C_{i,j} = \frac{1}{2} \frac{1}{MN - (M+N)/2} \sum_{m} \sum_{n} \mathbf{1}_{\frac{i}{255}}[g_{mn}] \left(\mathbf{1}_{\frac{j}{255}}[g_{m-1,n}] + \mathbf{1}_{\frac{j}{255}}[g_{m,n-1}] \right)$$
(30)

Ainsi si g_{mn} vaut $\frac{i}{255}$ et $g_{m-1,n}$ vaut $\frac{j}{255}$, alors ce couple de pixels (m,n) et (m-1,n) contribuent pour 1/2/(M-1)/(N-1) à la composante (i,j) de la matrice de coocurrence. Les composantes ainsi définies forment une matrice qui n'est pas nécessairement symétrique. La normalisation utilisée ici permet d'interprêter les composantes $C_{i,j}$ comme des probabilités qu'un couple de pixels aient telle transition. L'entropie du second ordre est définie par $H=-\frac{1}{2}\sum_{ij}p_{ij}\log_2(p_{ij})$

- 9. Calculez la matrice de coocurrence de l'image sans bruit.
- 10. Calculez l'entropie de l'image sans bruit.

Exercice 25 (18) On considère une image 256×256 définie par $f_{mn} = \frac{max(|m-128|,|n-128|)}{128}$.

- 1. Décrivez l'aspect visuel de cette image. On pourra d'abord considérer la matrice de taille 5×5 définie par $g_{mn} = max(|m-2|, |n-2|)$.
- 2. Calculez et représentez l'histogramme de cette image f.
- 3. On effectue un seuillage sur cette image f (c'est-à-dire que l'on quantifie cette image sur deux niveaux). Le seuil est fixé à 0.5. Décrivez l'image résultante.
- 4. Calculez le niveau de gris moyen de l'image obtenue.
- 5. Choisir le seuil de façon que l'image obtenue ait un niveau de gris moyen égale à 0.5.

On cherche maintenant à égaliser l'histogramme de cette image, c'est-à-dire à appliquer une transformation non-linéaire notée T sur les niveaux de gris de façon que le nouvel histogramme soit le plus plat possible.

- 6. Calculez l'histogramme cumulé, on rappelle que la somme des n premiers entiers est égale à $\frac{n(n+1)}{2}$.
- 7. Idéalement combien vaudrait un histogramme plat h^* et son histogramme cumulé C^* ?
- 8. La transformation non-linéaire doit vérifier $T(m) = \min\{l|C(m) \le C^*(l)\}$. Que vaut T?

Exercice 26 (23) On considère la transformation T suivante qui opère sur une image g_{mn} de taille $M \times N$, en niveaux de gris à valeurs sur [0,1]. $f_{m+1,n+1} = \frac{1}{3}f_{m,n+1} + \frac{1}{3}f_{m+1,n} + g_{m+1,n+1}$.

- 1. S'agit-il d'une transformation linéaire?
- 2. Montrez qu'elle est invariante par translation?
- 3. Cette transformation est donc un filtre linéaire, on cherche maintenant à déterminer le masque équivalent. On considère une image notée δ_{mn} qui vaut 1 en haut à gauche et zéro partout ailleurs. Calculez le résultat de cette transformation pour une image de taille 4×4 . On pose r = 1/3. On note f^{δ} le résultat de cette transformation.
- 4. En utilisant des notions de dénombrements montrez que $f_{m,n}^{\delta} = r^{m+n}C_{m+n}^n$. Ceci est le masque infini qui caractérise la transformation.
- 5. A partir de la formule de récurrence, montrez que la fonction de transfert de T est $T(z) = \frac{1}{1 rz_1^{-1} rz_2^{-1}}$.
- 6. En utilisant un développement en série de la fonction de transfert, retrouvez le masque infini.

Exercice 27 (30) Cet exercice montre la notion d'inférence bayésienne. Un objet uniforme est photographié dans un environnement bruité. Le bruit est un bruit blanc gaussien additif d'écart-type $\sigma=0.05$ et le fond de l'image est noir, l'objet a une teinte de $g^*=0.5$ avec une forme inconnue qui occupe une taille de $S^*=2500$ pixels au sein d'une image 256×256 . Comme l'objet est plus clair que le fond, il est naturel de penser que les zones claires donne une forte présomption de la présence de l'objet. La question que l'on se pose ici est, à partir de quelle intensité de gris, peut-on penser qu'il y a un objet. Autrement dit, nous allons seuiller l'image, mais on se demande comment choisir le seuil.

- 1. S'il n'y avait pas de bruit, quelle serait la probabilité pour un pixel de faire parti de l'objet? Cette probabilité est notée $P(I_{mn}^* = g^*)$. Quelle serait la probabilité $P(I_{mn}^* = 0)$ de ne pas faire parti de l'objet? Ces probabilités sont appelées probabilités a priori.
- 2. Les niveaux de gris des pixels constituent un processus aléatoire et en particulier le niveau de gris d'un pixel constitue une variable aléatoire. A priori il s'agit de quantités aléatoires discrètes mais ici pour simplifier nous considérons qu'il s'agit de quantités aléatoires continues. Si un pixel fait parti de l'objet, quelle est la densité de probabilité des niveaux de gris ? Il s'agit d'une densité de probabilité conditionnelle, elle est notée f[I_{m,n} = x|I^{*}_{mn} = g*](x). Si un pixel ne fait pas parti de l'objet, quelle est la densité de probabilité des niveaux de gris ? Elle est notée f[I_{m,n} = x|I^{*}_{mn} = 0](x).
- 3. Expliquez pourquoi la densité de probablité peut se décomposer ainsi

$$f(I_{mn} = x) = f(I_{mn} = x | I_{mn}^* = g^*) P(I_{mn}^* = g^*) + f(I_{mn} = x | I_{mn}^* = 0) P(I_{mn}^* = 0)$$

et pourquoi

$$f(I_{mn} = x \text{ et } I_{mn}^* = 1) = f(I_{mn} = x | I_{mn}^* = g^*) P(I_{mn}^* = g^*)$$

et finalement pourquoi

$$f(I_{mn}^* = g^* | I_{mn} = x) = \frac{f(I_{mn} = x | I_{mn}^* = g^*) P(I_{mn}^* = g^*)}{f(I_{mn} = x | I_{mn}^* = g^*) P(I_{mn}^* = g^*) + f(I_{mn} = x | I_{mn}^* = 0) P(I_{mn}^* = 0)}$$

Cette dernière égalité s'appelle la formule de Bayes.

- 4. On note r le ratio $\frac{f(I_{mn}^*=g^*|I_{mn}=x)}{f(I_{mn}^*=0|I_{mn}=x)}$. Montrez que r en fonction de x, on note cette fonction h. Montrez que $h(x) = \frac{S}{N^2 S} e^{\frac{2xg^* g^{*2}}{2\sigma^2}}$.
- 5. Montrez que la probabilité pour un pixel d'appartenir à l'objet en fonction du niveau de gris du pixel vaut $\frac{\mathcal{H}(x)}{1+h(x)}$. Montrez que cette fonction est croissante, et expliquez pourquoi cette croissance est cohérente avec la modélisation. On observe que pour $x=\frac{g^*}{2}$, alors la probabilité étudiée vaut $\frac{S}{N^2}$, expliquez cette valeur.
- 6. On décide de seuiller l'image à $s = h^{-1}(1)$. Calculez h^{-1} . Que peut-on affirmer à propos des pixels qui ont été seuillés (dont le niveau de gris est supérieur à s)? Expliquez pourquoi dans certaines applications on pourrait être amené à seuiller avec un seuil beaucoup plus faible : s = h(19).

Cours 30 (29)

Un ensemble peut se représenter au moyen d'une image binaire : l'ensemble des pixels valant 1. Un élément structurant est défini par un ensemble dont un élément est appelé centre. Un élément structurant est aussi un ensemble de translations, ce sont toutes les translations qui déplacent le centre de l'élément structurant en un des éléments de l'ensemble structurant. La dilatation (dilation) d'un ensemble A par un élément structurant B contenant un centre est noté $A \oplus B$ et est défini par l'image de A par l'ensemble des translations de B. L'érosion (erosion) d'un ensemble A par un élément structurant B contenant un centre est noté $A \oplus B$ et est défini par l'ensemble des points de A qui sont encore dans A après une des translations de B. L'ouverture (opening) de A par un élément structurant B est définie par une érosion suivie d'une dilatation de A par B. La fermeture (closing) de A par un élément structurant B est définie par une dilatation suivie par une érosion de A par B.

Exercice 28 (29) On considère l'image suivante :

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- 1. On considère l'élément structurant suivant $B = [1 \ 1 \ 1]$, le centre est indiqué en gras. Calculez l'image dilatée par B puis l'image érodée par B.
- 2. On considère l'élément structurant suivant $C = [1\, {f 1}; 0\, 1]$. Calculez l'image dilatée par B puis l'image érodée par C.
- 3. Calculez l'ouverture et la fermeture de I pour les éléments structurants B et C.
- 4. Il y a plusieurs définitions du gradient morphologique. Ici nous allons en considérer deux : l'intersection de l'image dilatée avec le complémentaire de l'image d'origine d'une part, et l'intersection de l'image d'origine avec le complémentaire de l'image érodée d'autre part. Calculez le résultat de ces deux gradients morphologiques pour l'élément structurant $D = [1 \ 1]$.

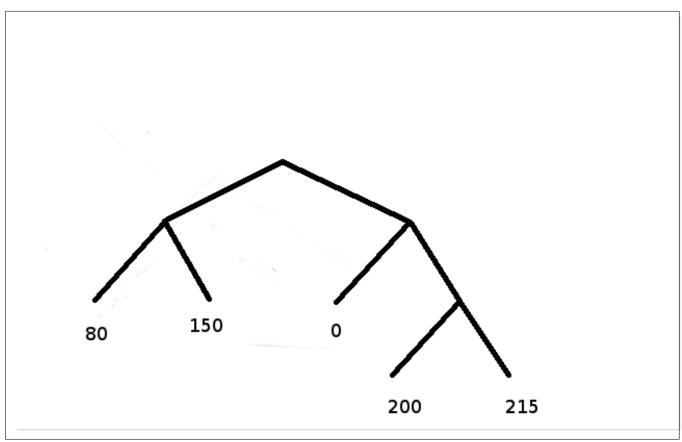


Figure 15: images de l'exercice 29

Exercice 29 (28) La chaîne suivante a été codée à l'aide du codage de Huffmann

1000100100001110010001110111111001000111001000100010010

On sait, à la lecture de l'entête que l'image de taille 5×5 et que le dictionnaire est le suivant :

$$\begin{array}{rcl}
0 & = & 10 \\
80 & = & 00 \\
150 & = & 01 \\
200 & = & 110 \\
215 & = & 111
\end{array}$$

- 1. Expliquez comment a été construit l'arbre présenté dans la figure 15 ?
- 2. Reconstruire l'image.
- 3. Calculez la longueur moyenne du code L, la longueur exacte du code L_e et l'entropie H. Montrez qu'on a effectivement $H \le L \le H + 1$. La figure 16 (p. 53) peut aider à faire le calcul numérique.
- 4. La règle d'Huffman qui conduit au choix des codes est qu'il faut coder les niveaux de gris les plus probables avec les séquences les moins les moins longues. Montrez que si on change le code en ne respectant pas cette règle par exemple en échangeant le code associé à 200 ou 215 à celui d'un des autres niveaux de gris, alors la longueur moyenne du code est augmentée de la différence de probabilités entre les deux codes.

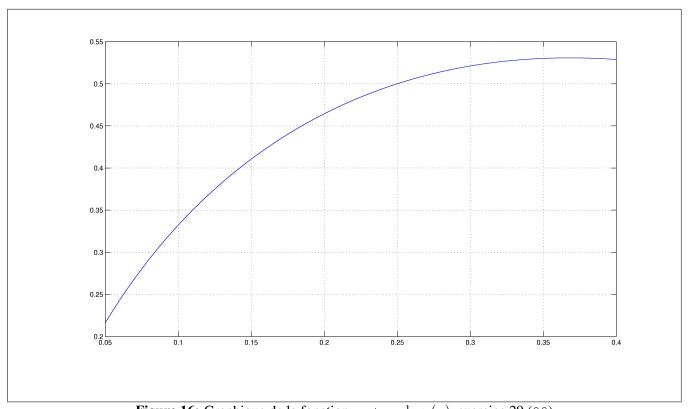


Figure 16: Graphique de la fonction $x \mapsto -x \log_2(x)$, exercice 29 (28)

Exercice 30 (27) Cet exercice comporte des difficultés similaires à celles rencontrées pour faire un sudoku, l'intérêt est qu'il permet de comprendre les difficultés et les nécessaires approximations à faire pour réaliser un algorithme de synthèse de texture. Les matrices de cooccurence sont en effet un outil permettant de décrire les textures non-déterministes.

On considère des images de taille 4×4 à valeurs dans $\{0, 1, 2, 3\}$.

1. Calculez la matrice de cooccurence pour les transitions verticales à un pas de l'image suivante :

$$I_1 = \left[\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{array} \right]$$

Dans la matrice de cooccurence, l'indice des colonnes indique le niveau de gris du pixel de départ et l'indice des lignes indique le niveau de gris du pixel à l'arrivée. Ici la matrice n'est pas normalisée et la somme des composantes est égale à l'ensemble des transitions verticales c'est-à-dire $3 \times 4 = 12$.

2. Proposez une matrices 4×4 à valeurs dans $\{0,1,2,3\}$ qui auraient la matrice de cooccurences suivantes pour des transitions verticales à un pas.

$$C_2 = \left[egin{array}{cccc} 3 & 0 & 0 & 0 \ 0 & 3 & 0 & 0 \ 0 & 0 & 3 & 0 \ 0 & 0 & 0 & 3 \end{array}
ight]$$

3. Montrez que les deux matrices de cooccurences ne correspondent à aucune matrice d'images.

On cherche maintenant à trouver une image dont la matrice de cooccurence est la suivante

$$C_5 = \left[egin{array}{cccc} 0 & 1 & 1 & 1 \ 1 & 0 & 1 & 1 \ 1 & 1 & 0 & 0 \ 1 & 1 & 2 & 0 \end{array}
ight]$$

Pour cela on propose de procéder par étapes.

4. Trouvez une image dont la matrice de coocurrence est :

$$C_6 = \left[\begin{array}{cccc} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array} \right]$$

Pour cela on cherchera parmi les images dont les colonnes et les lignes sont formées de chiffres différents. Expliquez pourquoi il est possible de choisir arbitrairement la première colonne et la première ligne sans craindre d'écarter une bonne solution.

5. On remarque que quand on fait circuler une colonne (i.e. transformation du quatre-uplet (a,b,c,d) en (b,c,d,a)), on supprime une transition et on en fait apparaître une autre. De même quand on remplace un chiffre par un autre on fait disparaître une transition et on en fait apparaître une autre. En combinant ces deux méthodes trouvez une image dont la matrice de cooccurence soit C_5 .

54