



DIRO
IFT 6150

EXAMEN THÉORIQUE

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle

[http : //www.iro.umontreal.ca/~mignotte/ift6150](http://www.iro.umontreal.ca/~mignotte/ift6150)

e-mail : mignotte@iro.umontreal.ca

Date : 13/12/2019

I	Transformée de Fourier (32 pts)
II	Spectres et Filtrage (38 pts)
III	Opération Ponctuelle (8 pts)
IV	Filtrage Spatial et Fréquentiel (16 pts)
V	Questions relatives aux TPs (36 pts)
Total	130 points

TOUS DOCUMENTS, CALCULATRICES ET CALCULATEURS PERSONNELS AUTORISÉS

I. Série de Fourier et Transformée de Fourier (32 pts)

Soit donc $f(x)$ un signal et $F(\nu)$ sa TF ou son spectre (*i.e.* ; $f(x) \rightleftharpoons F(\nu)$).

1. SÉRIE DE FOURIER VERSUS TRANSFORMÉE DE FOURIER

- (a) En utilisant la connaissance (vue en classe) de la TF de la fonction triangle ($\mathcal{F}[\Lambda(x)] = ([\sin \pi\nu]/[\pi\nu])^2$), trouver, en utilisant aussi le peigne de Dirac (et sa TF), le résultat que donnerait la TF de la fonction triangle périodique de période $T = 2$.

À cette fin, vous pouvez utiliser un raisonnement qualitatif ou mathématique (quantitatif) en sachant que plus vous serez précis, plus vous aurez la totalité des points de cette question.

<8 pts>

2. TRANSFORMÉE DE FOURIER

- (a) On aimerait calculer, dans le domaine des distributions, l'intégrale suivante :

$$f(x) = \int_{-\infty}^{+\infty} \cos(2\pi\nu x) d\nu$$

À cette fin, vous pouvez avoir un raisonnement qualitatif en vous demandant quel serait la TF du signal spatial qui serait composé de cette infinité de cosinus (pour les différentes fréquences réelles) et ensuite la distribution correspondante (à ce spectre) dans l'espace spatial ou avoir un raisonnement quantitatif en utilisant la TF et la TF inverse.

<6 pts>

- (b) Soit $f(x)$ un signal 1D et $F(\nu) = \mathcal{F}[f(x)]$ sa TF ou son spectre ($f(x) \rightleftharpoons F(\nu)$).

On réalise le filtrage fréquentiel suivant sur $F(\nu)$, le spectre d'un signal $f(x)$:

$$F_{filt}(\nu) = (2\pi j \nu)^{0.5} F(\nu)$$

Expliquer conceptuellement (qualitativement) ce que l'on va obtenir si on revient dans notre domaine spatial, *i.e.*, si on calcule maintenant : $\mathcal{F}^{-1}[F_{filt}(\nu)]$? (vous pouvez raisonner en terme de filtrage [type de filtre] et/ou de *dérivée particulière du signal*).

<6 pts>

- (c) Énumérer **toutes** les propriétés de $F(\nu)$, la TF de la fonction $f(x) = x \Lambda(x)$ (sans la calculer). Trouver ensuite cette TF (la calculer) en utilisant pas directement la définition de la TF utilisant la formule intégrale.

<6 pts>

- (d) En vous aidant de l'espace de Fourier, trouver à quoi est égale $\delta'(x) \cdot \cos(x)$.

<6 pts>

Réponse

1.

Raisonnement qualitatif : La périodisation de la fonction triangle $\Lambda(x)$, toutes les périodes $T = 2$, dans le domaine spatial se transforme dans le domaine spectral par la discrétisation du spectre de la fonction triangle (*i.e.* : $F(\nu) = [(\sin \pi\nu)/(\pi\nu)]^2$) pour toutes les fréquences n/T (où n est un entier). On aura donc, pour la TF de la fonction triangle qui se périodise, un spectre de raies (ou spectre discrétisé ou échantillonné) toutes les fréquences n/T (avec $n \in \mathbb{N}$).

Raisonnement quantitatif :

On peut reprendre le raisonnement de l'acétate 20 du chapitre *Transformée de Fourier* en remplaçant la fonction $f_0(x)$ par notre fonction triangle $f_0(x) = \Lambda(x)$. En effet, la fonction triangle périodique $f(x)$ peut se noter (avec $T = 2$) :

$$f(x) = \sum_{n=-\infty}^{n=+\infty} \Lambda(x - nT) = \Lambda(x) * \sum_{n=-\infty}^{n=+\infty} \delta(x - nT)$$

En appliquant la TF, et en appliquant la propriété $F(\nu) \cdot \delta(\nu - n/T) = F(n/T) \cdot \delta(\nu - n/T)$, on obtient :

$$F(\nu) = \left(\frac{\sin \pi \nu}{\pi \nu} \right)^2 \times \frac{1}{T} \sum_{n=-\infty}^{n=+\infty} \delta\left(\nu - \frac{n}{T}\right) = \sum_{n=-\infty}^{n=+\infty} \underbrace{\frac{1}{T} \left(\frac{\sin \pi n/T}{\pi n/T} \right)^2}_{A_n} \cdot \delta\left(\nu - \frac{n}{T}\right)$$

On obtient bien un spectre de raies équidistantes (caractéristique des spectres des signaux périodiques) et séparées dans notre cas de la fréquence $1/2$ (correspondant à la fréquence fondamentale) et pondéré par les coefficients A_n (qui sont aussi les coefficients A_n de la série de Fourier [associés au cosinus] pour $n \in \mathbb{N}^+$; les coefficients B_n étant nuls puisque la fonction $\Lambda(x)$ est paire).

NOTA : On peut vérifier que A_0 est bien la valeur moyenne ou composante continue du signal *fonction triangle périodique*; en effet $A_{n=0} = 1/2$ et correspond à la valeur moyenne de notre signal sur une période $T = 2$.

<8 pts>

2.(a)

Raisonnement qualitatif : La TF d'un signal cos pour une certaine fréquence est un spectre de deux raies dans le domaine spectral (réel puisque le signal cos est pair) d'amplitude $1/2$ et centré sur la fréquence positive et négative correspondant à la fréquence de ce signal cosinus. Puisque la TF est linéaire, la somme (ou l'intégrale) des spectres de ces différents cosinus (entre $\nu = 0$ et $+\infty$) va donc totalement remplir le domaine spectral réel d'une suite de raies spectrales d'amplitude $1/2$ pour toutes les fréquences continues (positives ou négatives) existantes et créer le signal constant $1/2$ dans le domaine spectral réel.

La somme (ou l'intégrale) des spectres de ces différents cosinus (entre $\nu = -\infty$ et 0) va faire la même chose et la somme de ces deux portions d'intégrales créera le signal constant 1 dans la partie réelle du spectre. Or on sait que la TF inverse de 1 (dans le domaine fréquentiel) est la distribution $\delta(x)$ dans le domaine spatial. Donc cette intégrale est égale à $\delta(x)$.

Raisonnement quantitatif :

$$f(x) = \int_{-\infty}^{+\infty} \cos(2\pi\nu x) d\nu = \int_{-\infty}^{+\infty} \cos(2\pi\nu x) + j \sin(2\pi\nu x) d\nu = \int_{-\infty}^{+\infty} 1 \exp(2\pi j \nu x) d\nu = \mathcal{F}^{-1}[1] = \delta(x)$$

<6 pts>

NOTA : On peut vérifier que le résultat $\delta(x)$ est bien réel comme l'intégrale et bien paire comme l'est toute somme de fonctions élémentaires paires.

2.(b)

En premier lieu, on peut dire que c'est un filtrage fréquentiel de la famille des filtres passe-haut avec une réponse fréquentielle $(2\pi j \nu)^{0.5} \propto 2.5 \nu^{0.5}$ qui va atténuer et annuler les basses fréquences (pour $\nu = 0$ annuler) et amplifier les hautes fréquences (pour $\nu = 0.5$, fréquence numérique la plus grande, les amplifier de ≈ 1.7).

Si maintenant on se réfère à la définition de dérivation de la TF, ce filtrage fréquentiel va nous permettre d'obtenir, dans notre domaine spatial, la dérivée d'ordre (non entière ou flottante) $1/2$ du signal $f(x)$ que l'on pourrait modéliser ou noter mathématiquement de la façon suivante :

$$\mathcal{F}^{-1}[(2\pi j \nu)^{0.5} F(\nu)] = \frac{\partial^{1/2} f(x)}{\partial x^{1/2}}$$

Si on voulait retrouver l'opération spatiale qui nous conduise à cette dérivée fractionnaire d'ordre 0.5 et bien cela correspondrait à la convolution de l'image $f(x, y)$ suivant $f(x, y) * \mathcal{F}^{-1}[(2\pi j \nu)^{0.5}]$ mais il n'est pas évident (et peut-être impossible!) de trouver : $\mathcal{F}^{-1}[(2\pi j \nu)^{0.5}]$ (mais en fait cela n'a aucune importance car on peut réaliser facilement cette opération dans le domaine fréquentiel).

<6 pts>

2.(c)

La fonction $f(x) = x \Lambda(x)$ est le produit d'une fonction impaire et paire, elle est donc impaire (et réelle) donc sa transformée $F(\nu)$ sera imaginaire pure et impaire. De plus, comme la fonction $f(x)$ est non périodique, son spectre sera continu.

En utilisant $\mathcal{F}[x] = \frac{j}{2\pi} \delta'(\nu)$, la TF de la fonction Triangle vue en cours, et la propriété de convolution par le Dirac dérivée ($\delta'(\nu) * F(\nu) = F'(\nu)$), on a donc :

$$x \Lambda(x) \Rightarrow \frac{j}{2\pi} \delta'(\nu) * \left(\frac{\sin(\pi\nu)}{\pi\nu} \right)^2 = \frac{j}{2\pi} \cdot \frac{2 \sin(\pi\nu) (\pi\nu \cos(\pi\nu) - \sin(\pi\nu))}{\pi^2 \nu^2} = j \frac{\sin(\pi\nu) (\pi\nu \cos(\pi\nu) - \sin(\pi\nu))}{\pi^3 \nu^2}$$

Dont le spectre est bien imaginaire pure, impaire et continu.

<6 pts>

2.(d)

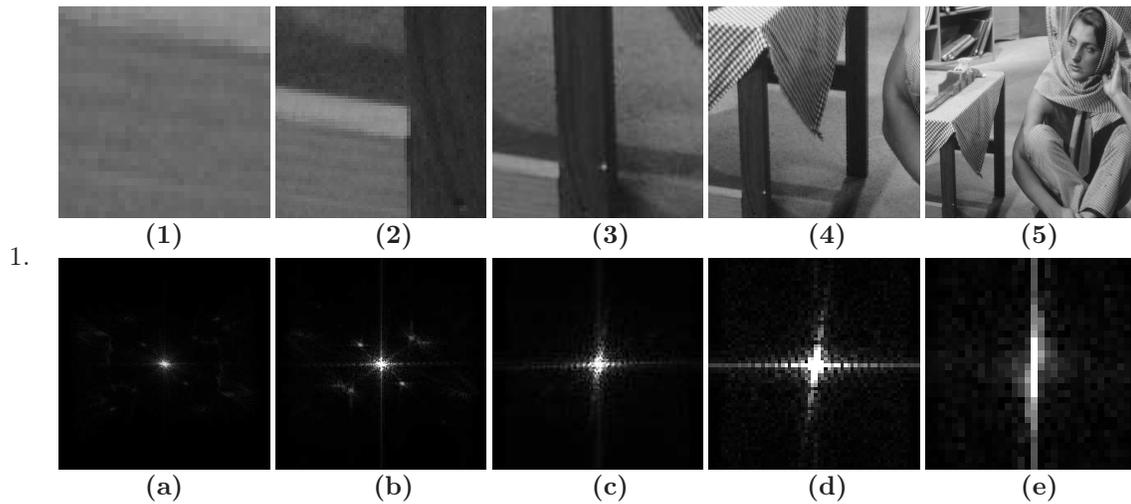
En utilisant la TF de la fonction δ' et la TF de la fonction cosinus vue en cours, et la propriété de convolution par le Dirac dérivée, on a :

$$\begin{aligned} \delta'(x) \cos(x) &\Rightarrow (2\pi j \nu) * \frac{1}{2} (\delta(\nu - (1/2\pi)) + \delta(\nu + (1/2\pi))) \\ &= [\pi j (\nu - (1/2\pi))] + [\pi j (\nu + (1/2\pi))] \\ &= 2\pi j \nu \end{aligned}$$

Donc $\delta'(x) \cos(x) = \delta'(x)$

<6 pts>

II. Spectres et Filtrage (38 pts)



Les images (1) à (5) de la Figure ci-dessous représentent le module du spectre de Fourier (calculé par la FFT puis recentré au centre de l'image et après avoir rehaussé légèrement les hautes fréquences comme en TP) des différentes imageries (1) à (5) prises sur l'image BARBARA de taille 512×512 (à partir du bord inférieur gauche et avec une taille chaque fois deux fois plus grande).

Indiquer, pour chacun des spectres, à quelle image il correspond. Justifier qualitativement chaque réponse par les caractéristiques présentes dans l'image et dans son spectre associé. Vous pouvez aussi raisonner par élimination.

<20 pts>

2. Soit $f(x, y)$ une image comme celle ci-dessous de Joseph Fourier (image ci-dessous) et $F(\nu, u) = \mathcal{F}[f(x, y)]$ sa TF ou son spectre (*i.e.*, son module recentré au centre de l'image) ($f(x, y) \Leftrightarrow F(\nu, u)$).

(a) On réalise le filtrage spatial suivant sur $f(x, y)$ qui consiste à mettre à zéro tous les niveaux de gris associés aux pixels situés une ligne sur deux et une colonne sur deux. Trouver quel sera le spectre de cette nouvelle image. Justifiez le mieux possible votre réponse.

<10 pts>



(b) À partir de l'image précédente *i.e.*, $f(x, y)$ pour laquelle on a mis une ligne et colonne sur deux à zéro), Trouver quel doit être l'opération fréquentielle qui permettrait de, après une TF inverse, de retrouver l'image $f(x, y)$ initiale non dégradée. Indiquez d'une part quelle serait la condition pour que cela soit possible et l'opération fréquentielle qui permettrait de le faire. Dans cette opération, n'oubliez pas aussi d'intégrer le théorème de Parseval (conservation de l'énergie dans les deux espaces).

<8 pts>

Réponse

1.

On pourrait facilement, tout de suite, classer $[1 - e]$ et $[2 - d]$ car on voit que ses images ont le même nombre de pixels que leur transformée respective. Si on a de bons yeux, puisqu'il y a le même nombre de

pixels entre le signal et sa TFD on a aussi immédiatement $[3 - c]$, $[4 - b]$ et $[5 - a]$. Mais même sans cela on peut arriver au bon classement.

Le plus simple est de procéder par élimination en allant du plus évident au moins évident :

- $[1 - e]$:

Belle transition verticale indiquant une belle discontinuité horizontale dans le domaine spatial. La seule possibilité est l'image (1) qui présente une transition horizontale (la première ligne est claire et la dernière ligne est plus sombre) laquelle est due à la périodisation implicite de l'image (en ligne et en colonne) créée par la TFD (ou plus précisément la version discrétisée de la TF). Elle est très visible car l'image est petite et comporte peu de pixels (quand l'image comporte peu de pixels, les transitions de bords occupent un pourcentage important du nombre total de pixels de l'image [cf. examen de l'année précédente]).

- $[2 - d]$:

Il existe deux directions de discontinuité dans cette image (verticale et horizontale légèrement descendante) créant deux discontinuités dans la direction opposée dans le domaine spectral.

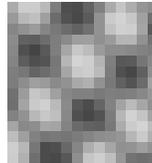
- $[5 - a]$:

L'image possède plein de textures différentes et toute cette information est résumée avec le pouvoir compressif du calcul de la TF par une tache au centre de l'image.

- $[4 - b]$:

Même raisonnement que précédemment si ce n'est que l'on aperçoit nettement dans cette image une texture avec un motif périodique (la nappe de la table) créant les 4 pics d'intensité autour de la tache centrale comme si le spectre résultant était discrétisé par un peigne de Dirac ; donc indiquant qu'un texton se répète périodiquement dans le domaine spatial dans deux directions opposées correspondant aux deux directions de la nappe. Ce texton est l'élément périodique de la nappe, c'est à dire le motif carré clair-sombre.

NOTA : On peut même dire que la fréquence numérique associée à ces pics est autour de 0.15 correspondant à un motif périodique qui se répète dans l'image spatiale tous les $0.10^{-1} \approx 9$ ou 10 pixels ce qui est confirmé par l'image et par l'extrait de celle-ci que je donne à la suite de ce commentaire.



- $[3 - c]$:

Deux directions de discontinuités affaiblies par quelques autres discontinuités de direction différentes.

<20 pts>

2.(a)

▷ Qualitativement, l'opération faite dans le domaine spatial est une discrétisation d'un échantillon sur deux (sur les lignes et colonne) de l'image. Donc, puisque la discrétisation dans un espace équivaut à la périodisation dans l'autre espace, on aura une périodisation du motif spectral dans l'espace fréquentiel suivant les fréquences lignes et fréquences colonnes (toutes les moitiés d'images).

▷ Mathématiquement supprimer une ligne et colonne sur deux, correspond sur l'image $f(x, y)$ à l'opération suivante (en considérant x (la colonne) et y (la ligne) et la correspondance $x \rightleftharpoons u$, et $y \rightleftharpoons \nu$:

$$f(x, y) \cdot \left\{ \text{III}_{T_x=2, T_y=2}(x, y) \right\} \stackrel{\text{TFD}}{\rightleftharpoons} F(u, \nu) * \left\{ \text{III}_{T_u=\frac{1}{2}, T_\nu=\frac{1}{2}}(u, \nu) \right\}$$

Le motif spectral (*i.e.*, le spectre de $f(x, y)$) va donc se répéter toutes les fréquences numériques $1/T = 1/2$ en coordonnée spectrale ligne et spectrale colonne, c'est-à-dire toutes les moitiés de l'image en u (fréquences spatiales colonnes) et (fréquences spatiales lignes) on obtiendra :



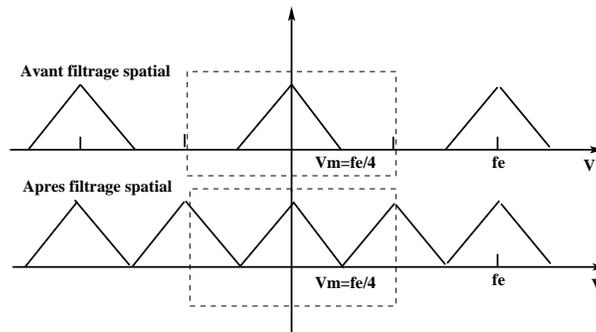
<10 pts>

2.(b)

Pour revenir à l'image initiale $f(x, y)$, il faut donc retrouver le spectre de $f(x, y)$ non dupliqué, donc supprimer cette duplication de $F(u, \nu)$ en ligne et colonne. L'opération fréquentielle que l'on doit donc faire est un filtre passe-bas idéal avec une fréquence de coupure idéalement placée à la fréquence numérique en ligne et colonne de $u = \nu = 0.25$.

Pour que ce filtre passe-bas, retrouve parfaitement (c-a-d sans altération) le spectre de $f(x, y)$, (c-a-d $F(u, \nu)$), il faut que la fréquence maximale du signal en ligne et colonne soit inférieure à la fréquence numérique $u = \nu = 0.25$.

Donc, pour l'image originale $f(x, y)$, il faut que l'on échantillonne à au moins 4 fois la fréquence maximale contenue dans l'image pour que cette opération fréquentielle (filtrage passe-bas idéal sans dégradation du motif spectral initial) soit possible (cf. Fig.).



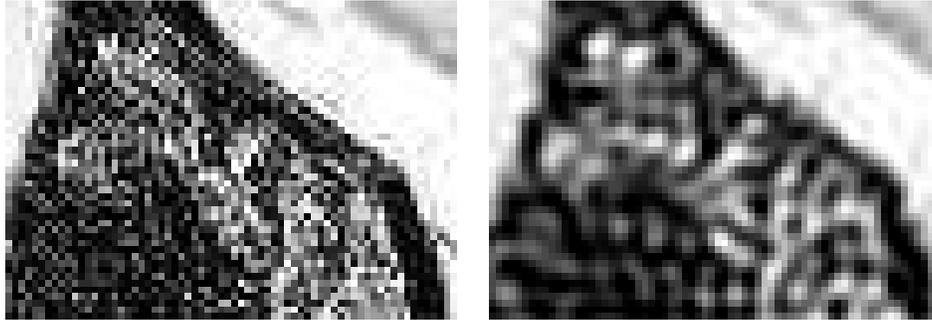
De plus, le fait d'annuler un échantillon sur deux de $f(x, y)$ va réduire l'énergie du signal par deux. Une fois que l'on aura fait le filtrage passe-bas et la TF inverse, on aura le même signal qu'au départ mais avec une amplitude deux fois moins grande (donc une énergie 4 fois moins grande). On devra donc multiplier par quatre l'intensité de l'image filtrée obtenue.



<8 pts>

NOTA : Si on regarde de près la reconstruction (ou restauration) obtenue, on s'aperçoit que la condition énoncée précédemment ("il faut que l'on échantillonne à au moins 4 fois la fréquence maximale contenue

dans l'image pour que cette opération fréquentielle"), n'a sans doute pas été respectée car il y a certaines zones hautes fréquences où la reconstruction est légèrement différente et moins détaillée que celle de l'image originale.



Portion très agrandie de l'image initiale et de l'image finale.

III. Opération Ponctuelle (8 pts)

1. Pour améliorer la visibilité des détails d'une image médicale à rayons X et permettre ainsi plus facilement son interprétation, vous conseillez au médecin d'alterner entre égalisations d'histogramme et étirement d'histogramme pour améliorer le contraste et donc la visibilité des détails lorsque les détails intéressants de l'image sont représentés dans une certaine gamme ou dynamique de niveaux de gris. Parmi les conseils que vous lui donnez, vous lui dites de revenir à l'image originale après une égalisation d'histogramme et avant de faire un étirement linéaire de l'histogramme (avec une possible saturation) sur une certaine zone de niveaux de gris et de ne surtout pas faire un étirement d'histogramme directement après une égalisation d'histogramme. Expliquez pourquoi en détaillant un peu votre réponse.

<8 pts>

Réponse

1.

Une égalisation d'histogramme permet d'améliorer la visibilité des détails de l'image uniformément dans toutes les gammes de niveaux de gris possible et un étirement d'histogramme (avec possible saturation) dans une certaine zone de niveaux de gris. Ses deux opérations permettent de mieux apprécier les détails de l'image globalement et localement mais peuvent perdre de l'information. Des détails, représentés par de légères variations de niveaux de gris, peuvent se transformer après ces deux types d'opération en zones de région de niveaux de gris homogène (deux barres de l'histogramme se confondent en une seule barre) et seront donc inéluctablement perdus.

Si on passe directement d'une égalisation d'histogramme à un étirement local avec saturation, on perdra toutes les informations dû à l'égalisation d'histogramme et toutes les informations (détails) dus à l'étirement d'histogramme avec saturation et si on répète, après, différents étirements locaux avec saturation dans différentes gammes de niveaux de gris, on éliminera de plus en plus de détails de l'image. C'est pourquoi il faut revenir à l'image originale (avec son histogramme brute) avant de faire une égalisation d'histogramme ou avant tout étirement d'histogramme avec saturation.

<8 pts>

IV. Filtrage Fréquentiel (16 pts)

1. Soit le commentaire et la commande Matlab suivante, extraite d'un programme Matlab existant et disponible sur le Web (et que l'on peut retrouver sur le Web en mettant cette commande avec le commentaire associé dans un moteur de recherche) :

```
% Construct a low-pass filter that is as large as possible, yet falls
% away to zero at the boundaries.
lp = lowpassfilter([rows,cols],.4,10); % Radius .4, 'sharpness' 10
```

On aimerait programmer en C le filtre en question. Je vous aide ; le *Radius* fait référence à la fréquence de coupure du filtre (en terme de fréquence numérique) et la *sharpness* fait référence au fait que le filtre en question n'est pas idéal (et donc au filtre de butterworth vu en cours). En vous aidant du cours, trouver les 5 ou 6 lignes de codes permettant de construire ce filtre dans un tableau 2D : *Matlp[][]* (en considérant le centre de ce filtre comme étant le centre du tableau 2D pour simplifier l'implémentation).

<8 pts>

2. Voici l'image BARBARA de taille 512×512 et une version croppée de la nappe qui se trouve au milieu et à gauche de l'image.



On aimerait faire un filtrage fréquentiel qui floute (*i.e.*, qui remplace) le motif carré blanc-noir de la nappe (qui se répète) par une zone quasi uniforme. Définir le type de filtrage fréquentiel et les paramètres associés à celui-ci (en fréquence numérique ligne et fréquence numérique colonne).

<8 pts>

Réponse

1.

Il s'agit bien sûr du filtre passe-bas non-idéal donc du filtre de Butterworth vu en cours avec $D_0 = 0.4$ et $2n = 10$ (sharpness). Il faut faire en sorte que $D(u, \nu)$ varie (horizontalement et verticalement) entre les fréquences numériques 0 et 0.5.

```

MidLgth=length/2;
MidWdth=width/2;
Radius=0.4;
Sharpness=10;

for(i=0;i<length;i++) for(j=0;j<width;j++)
{
dist=sqrt(CARRE((i-MidLgth)/(float)length)+CARRE((j-MidWdth)/(float)width));
Matlp[i][j]=(1.0/(1.0+powf((dist/Radius),Sharpness)));
}

```

Créera le filtre passe-bas au centre de l'image.

NOTA : N'oubliez pas que la fréquence (numérique) doit être de 0.5 aux extrémités horizontales et verticale de l'image.

<8 pts>

2.

Le motif carré blanc-noir se répète toutes les (environ) $T_x = 10$ colonnes et $T_y = 10$ lignes et en diagonalement droite de $T_{dd} = 5$ pixels et en diagonalement gauche de $T_{dd} = 10$ pixels.

Donc il nous faudra un filtrage coupe-bande selon une ellipse associée avec une fréquence numérique en ligne et colonne et diagonalement gauche de $0.1 = (1/10)$ et une fréquence numérique en diagonalement droite de $0.2 = (1/5)$ et avec une largeur de bande raisonnable (modélisant l'incertitude que nous avons en T_x et T_y) du type 0.025 ou 0.05.

<8 pts>

V. Questions Relatives aux TPs (36 pts)

1. Dans le TP numéro 1 **TRANSFORMÉE DE FOURIER**.

- (a) Dans la question C. LA CONVOLUTION, un étudiant a programmé l'auto-convolution du carré (qu'il a décidé de mettre sur le côté gauche) de la façon suivante (avec MatI initialement remplie de 0 ET MatR de dimension lgth \times wdth) :

```

for(i=0;i<lgth;i++) for(j=0;j<wdth;j++) MatR[i][j]=255
FTDD(MatR,MatI,lgth,wdth);
CarreMatrix(MatR,MatI,lgth,wdth);
IFTDD(MatR,MatI,lgth,wdth);
...etc...

```

Avec :

```

CarreMatrix(float** MatR, float** MatI, int lgth, int wdth)
{
for(i=0;i<lgth;i++) for(j=0;j<wdth;j++)
. {
. MatR[i][j]=MatR[i][j]*MatR[i][j]-MatI[i][j]*MatI[i][j];
. MatI[i][j]=2*MatR[i][j]*MatI[i][j];
. }
}
}

```

Identifier où se trouve l'erreur dans la fonction CarreMatrix() ?
Réécrivez la fonction CarreMatrix() sans erreur.

<7 pts>

2. Dans le TP numéro 3 **RESTAURATION**.

- (a) Dans la question A.(1) on réalise une déconvolution de l'image en utilisant un algorithme itératif (utilisant la convolution réalisée dans le domaine fréquentiel) de descente du gradient ("one step gradient") en partant initialement de l'image floue (et avec une estimation au sens des moindres carrés ou au sens de la minimisation de la fonction de coût donnée en Eq. (2)). Quel serait le système d'équations équivalent qui permettrait de trouver cette image déconvoluée au sens de ce même critère et pourquoi n'utilise-t-on pas cette stratégie de résolution ?

<4 pts>

3. Dans le TP numéro 4 **TOMOGRAPHIE**.

- (a) Indiquez tous les éléments (matériel et logiciel) qui permettent à une reconstruction tomographique (pour les deux méthodes vues en cours) d'obtenir une image reconstruite avec une meilleure résolution.

<4 pts>

4. Dans le TP numéro 5 **SEGMENTATION MARKOVIENNE**.

- (a) Expliquer quels sont les tests qui nous permettent de savoir si on a programmé correctement un ICM ; *i.e.*, comment doit-il se comporter normalement et comment évolue le résultat de la segmentation pour différentes valeurs de α ($\alpha = 0$ ou pour des valeurs de plus en plus grandes et des segmentations initiales différentes).

<4 pts>

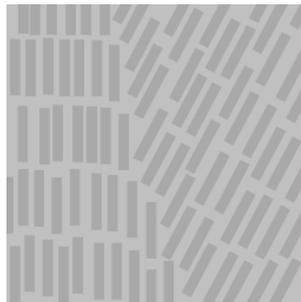
- (b) Après avoir implémenté l'ICM, Expliquer quels sont les tests qui nous permettent de savoir si on a programmé correctement un Recuit Simulé ; *i.e.*, comment doit-il se comporter normalement et comparativement à l'ICM et est-ce que l'on a toujours une optimisation plus efficace qu'un ICM ? Dans quel cas ? discuter aussi de la sensibilité de segmentation initiale).

<4 pts>

- (c) Au lieu de faire, à chaque itération de l'ICM, un balayage lexicographique des pixels, de gauche à droite et de haut en bas de l'image (symbolisé par les deux boucles `for(;;) for(;;)` itérant tous les pixels en lignes et ensuite en colonne), on décide de faire un balayage aléatoire de l'image qui prendrait n fois (n étant le nombre de pixels de l'image) un pixel aléatoirement dans l'image puis déciderait de son étiquette de classe. Obtiendrait-on le même résultat qu'un balayage lexicographique ? expliquer pourquoi et donner les quelques lignes de code qui permettraient de le faire algorithmiquement et qui se substitueraient au deux boucles `for(;;)` que vous avez utilisé dans le cadre d'un balayage lexicographique classique.

<6 pts>

- (d) Soit l'image ci-dessous pour laquelle on peut voir deux régions homogènes au sens de la texture ; la première région dans laquelle des sortes de bâtonnets sont orientés plutôt verticalement et une deuxième région où ceux-ci sont orientés plutôt dans le sens diagonal-droit.



On aimerait faire une segmentation en deux classes de cette image ; soit la classe d'étiquette *noire* pour la première région et l'étiquette *blanche* pour la deuxième.

On aimerait aussi utiliser l'algorithme utilisé et programmé au dernier TP. Dans cette optique, où on désire changer le moins de chose possible comment et/ou quelle partie et de quelle façon devrait-on changer cet algorithme pour réaliser cette opération de segmentation ?

<7 pts>

Réponse

1.(a)

Pour chaque pixel avec une valeur de spectre complexe du style $a + jb$, le résultat de la multiplication, pixel à pixel, de ce spectre, nous donne pour chaque pixel de l'image $(a + jb)^2 = (a^2 + b^2) + j(2ab)$. C'est-à-dire un complexe de partie réelle $(a^2 + b^2)$ que l'on doit mettre dans MatR et $2ab$ dans MatI.

Dans le code qu'on nous donne, la partie réelle est correcte, mais MatI est remplie, non pas de $2ab$ ou $2\text{MatR}[i][j] \cdot \text{MatI}[i][j]$ mais de $2(\text{MatR}[i][j]^2 + \text{MatI}[i][j]^2) \cdot \text{MatI}[i][j]$ (ou de $2(a^2 + b^2) \cdot b$!)

Donc le bon code est :

```
CarreMatrix(float** MatR, float** MatI, int lgth, int wdth)
{
float pr,pi;
for(i=0;i<lgth;i++) for(j=0;j<wdth;j++)
. {
. pr=MatR[i][j]*MatR[i][j]-MatI[i][j]*MatI[i][j];
. pi=2*MatR[i][j]*MatI[i][j];
. MatR[i][j]=pr;
. MatI[i][j]=pi; . }
}
```

<7 pts>

2.(a)

Si $g(x, y)$ et $f(x, y)$ correspond au niveaux de gris de l'image floue et l'image que l'on cherche (débarassée de son floue), on a pour une image carrée de taille 256×256 ($sz=256$), et un floue 5×5 uniforme, un système de 65536 équations (demandant beaucoup d'opérations) du style (x :ligne et y :colonne) :

$$\forall (x, y) \in [0, \dots, 255] \times [0, \dots, 255] \quad g(x, y) = (1/25) \sum_{m=-2}^{m=+2} \sum_{n=-2}^{n=+2} f[(x+m+sz)\%sz, (y+n+sz)\%sz]$$

et en considérant l'image f toroidale (avec % la fonction modulo et expliquant ainsi le $\dots + sz$)%sz ajouté en indice colonne et ligne).

On pourrait aussi considérer le système de $256 \times 256 = 65536$ équations donné en exprimant pour chaque pixel $\forall (x, y) \in [0, \dots, 255] \times [0, \dots, 255]$ (pour une image carrée de taille 256×256) résultant de l'équation d'erreur : $\frac{\partial^2}{\partial x \partial y} \|g(x, y) - h(x, y) * f(x, y)\|^2 = 0$ mais cela créerait aussi un système demandant beaucoup trop de calcul.

<4 pts>

3.(a)

Pour augmenter la résolution de l'image, il faudrait augmenter matériellement : Le nombre de projections, le nombre de rayons (ou le nombre de détecteur qui déterminent la taille des coupes projectives en terme

de nombre de pixels), le rapport signal sur bruit des capteurs, et logiquement ; la qualité de l'interpolation utilisée dans la rétroprojection filtrée ou dans l'algorithme de tranche centrale, un sinogramme dont la largeur (ou taille des projections) est plus grande que $(sz \times \sqrt{2})$ avec sz la taille carré de l'image) pour éviter l'artefact circulaire du à l'absence d'information sur les bords lors de la simulation du tomographe (ou rotation de l'image) et filtré avec un filtre passe-haut encore plus robuste au bruit.

<4 pts>

4.(a)

L'énergie doit toujours décroître (car c'est une descente du gradient et que l'on garde, d'itération en itération, la configuration de segmentation qui fait décroître la fonction d'énergie). Au début la décroissance d'énergie est grande et celle-ci diminue le long des itérations jusqu'à stagner au bout de quelques itérations ; on doit donc observer donc une courbe de décroissance d'énergie qui décroît un peu comme une fonction exponentielle. Pour une valeur $\alpha = 0$, on a une convergence de l'algorithme au bout d'une itération, et on retrouve la segmentation au sens du Maximum de Vraisemblance (MV), et lorsqu'on fait augmenter la valeur de $\alpha = 0$, la régularisation de la segmentation augmente ; c'est-à-dire que l'on élimine de plus en plus de petite zone de régions homogène. De plus, la solution finale dépend fortement de la segmentation initiale.

<4 pts>

4.(b)

Les remontées d'énergie sont permises d'itération en itération mais globalement l'énergie doit diminuer. Pour une décroissance de température assez lente, on devrait arriver à une énergie plus faible que celle atteinte par l'algorithme ICM, c'est-à-dire arriver à une solution de segmentation plus proche du MAP. Si ce n'est pas le cas alors on a pris une décroissance en température trop forte. Plus on prend une décroissance de température lente (en passant par exemple d'une décroissance du type $T^{[k]} = (0.999)^k T_o$ à $T^{[k]} = (0.9999)^k T_o$, plus on obtient un minima (de fonction d'énergie) plus faible mais au prix d'un plus grand nombre d'itérations. Le RS ne dépend pas de la segmentation initiale qui peut être aléatoire. Avec une température très proche du zéro (dernières itérations du RS), le RS est en fait un ICM et se comporte comme telle avec une fonction d'énergie qui ne peut que décroître.

<4 pts>

4.(c)

On n'obtiendrait pas la même chose car la descente de gradient se ferait en utilisant un autre chemin (le chemin est donné par la suite des pixels/variable utilisé). On tomberait donc dans un autre *minima* local qui serait peut-être plus intéressant ou moins intéressant que celui donné par un balayage classique lexicographique.

Une autre façon de raisonner est de dire que le résultat sera forcément différent car avec un balayage aléatoire, certains pixels serait balayé plus que certains autres (bien que plus on fait de balayage aléatoire, moins cette assertion est vraie ...) et comme les pixels modifiés à une position donnée influencent ensuite les pixels voisins qui influencent ensuite leur voisins, la suite des pixels est donc importante dans le résultat final.

Pour l'implémentation, il suffirait simplement de remplacer les deux boucles `for(;;)` que l'on a mis en commentaire en dessous par les quatre lignes suivantes :

```

for(m=0;m<(lgth*wdth);m++)
%-for(i=0;i<lgth;i++) for(j=0;j<wdth;j++) %-On met cette ligne en commentaire
{
n=randomize()*((lgth*wdth)-1);
i=(int)(n/wdth);
j=n-(i*wdth);
... ..
...etc...
}

```

À cette fin, on utilise, la fonction `randomize()` (ou `rand()/RAND_MAX`) que l'on a utilisée en TP pour générer un nombre aléatoire compris entre 0 et 1.

<6 pts>

NOTA : On peut faire + simple en mettant au début de {
`i=(int)(randomize()*(float)lgth); j=(int)(randomize()*(float)wdth);`

4.(d)

En fait dans le dernier TP, on avait une segmentation deux classes qui était faite de telle façon que les pixels les plus sombres (ou plus précisément dont la répartition ou distribution avait la moyenne la plus faible) appartenait à la classe 0 et les pixels les plus clairs (distribution dont la moyenne était la plus grande) devaient appartenir à la classe 1. Ici, on aimerait que les pixels appartenant à la région où les bâtonnets sont verticaux appartiennent à la classe 0 et les autres à la classe 1.

Le plus simple consiste à trouver un descripteur qui transforme l'image texturale en une image en niveaux de gris dans laquelle les pixels appartenant à la région où les bâtonnets sont verticaux soient associés à des niveaux de gris faibles et fort concernant les autres. Si on arrive à cela, on pourra utiliser ensuite l'algorithme ICM programmé en TP tel quel.

Une solution consiste à binariser l'image texturale (segmentation en deux classes à partir d'un seuil $\tau = \text{minGL} + \text{maxGL}/2$ où *minGL* et *maxGL* sont le min et max en niveaux de gris de l'image originale. Puis calcul des contours binaires de cette image binaire (calcul du module du gradient et lorsque celui-ci n'est pas nul, on met à 255 le pixel) puis pour chaque petite fenêtre centré sur le pixel considéré de taille par exemple 31×31 et pour lequel on aurait calculé la valeur moyenne de l'orientation de ces contours. Cf. images ci-dessous où l'on voit l'image texturée, l'image (binaire) des contours de celle-ci, la conversion en valeurs moyenne de l'orientation de ces contours sur une fenêtre 31×31 (image recalée entre 0 et 255), la segmentation binaire obtenue et l'image de la segmentation surimposée sur l'image de départ).

<7 pts>

