



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

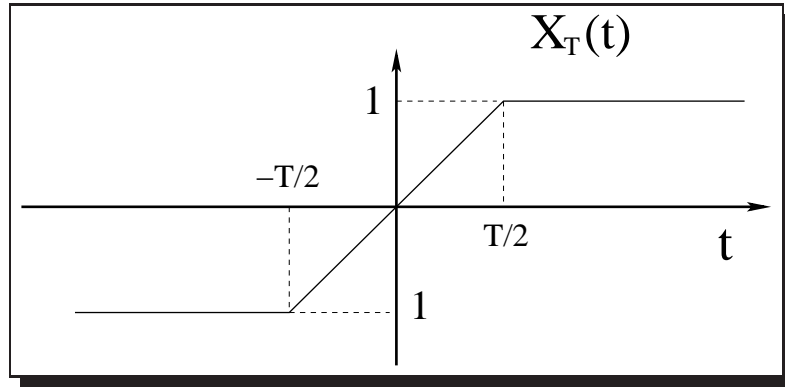
Date : 09/12/2014

I	Transformée de Fourier (39 pts)
II	Opérations Ponctuelles (18 pts)
III	Filtrage (35 pts)
IV	Misc. (36 pts)
Total	128 points

TOUS DOCUMENTS, CALCULATRICES ET CALCULATEURS PERSONNELS AUTORISÉS

I. Transformée de Fourier (39 pts)

1. Soit le signal temporel $x(t) = 4 \sin^2(t)$ et la correspondance $t \xrightarrow{\mathcal{F}} \nu$.
 - (a) Calculer $X(\nu)$, le spectre (ou la transformée de Fourier) du signal $x(t)$.
<4 pts>
 - (b) Calculer $X_e(\nu)$, la transformée de Fourier du signal $x(t)$ échantillonné à $\nu_e = 4/\pi$.
Est-ce que cette fréquence d'échantillonnage respecte la condition de Shannon?
<5 pts>
2. (a) Soit la fonction temporelle $x_T(t)$ représentée ci-dessous :



Soit aussi $X_T(\nu) = \mathcal{F}[x_T(t)]$, i.e., la transformée de Fourier (TF) de $x_T(t)$. Sans calcul, quelles seraient (toutes) les propriétés intéressantes de $X_T(\nu)$ que l'on pourrait déduire des propriétés de $x_T(t)$ et des propriétés de la TF.

- (a) <4 pts>
 - (b) Représenter l'allure de $x'_T(t)$, la dérivée de la fonction $x_T(t)$ et exprimer analytiquement $x'_T(t)$ à partir d'une fonction connue pour laquelle on connaît sa TF.
<4 pts>
 - (c) En déduire finalement, à partir d'une propriété de la TF, $X_T(\nu)$, la TF de $x_T(t)$.
<4 pts>
 - (d) Que devient (i.e., quel est le nom de la fonction) $x_T(t)$ quand $\lim_{T \rightarrow 0}$. Déduire des questions précédentes la TF de cette fonction (i.e., $\mathcal{F}[x_{T \rightarrow 0}(t)]$) et rappeler pourquoi on n'aurait pas pu calculer la TF de cette fonction en utilisant directement la définition de la TF utilisant la formule intégrale.
<4 pts>
 - (e) Déduire de la question précédente la TF de la fonction $\frac{j}{t}$ (où j est l'imaginaire pur).
<4 pts>
3. (a) Considérons maintenant des signaux spatiaux et montrer que pour n paire :

$$\frac{\partial^n}{\partial x^n} \{f(x) * g(x)\} = \left\{ \frac{\partial^{n/2}}{\partial x^{n/2}} f(x) \right\} * \left\{ \frac{\partial^{n/2}}{\partial x^{n/2}} g(x) \right\} \quad (1)$$

<5 pts>

- (b) La relation précédente (Eq. (1)) n'est pas abstraite et inutile pour un cours de traitement d'images. En effet, dans le chapitre **FILTRAGE SPATIAL**, nous avons présenté le filtre passe-bande (filtre de Marr-Hildreth) sur une image comme le résultat, sur celle-ci, de deux opérations. La première

est la convolution de l'image par un filtre Gaussien et la deuxième comme la convolution du résultat obtenu par un filtre Laplacien (modélisant une dérivée seconde). Avec la connaissance de la relation précédente, on aurait pu obtenir ce même résultat d'une autre façon. Laquelle?
 <5 pts>

Réponse

1.(a)

$$x(t) = 4 \sin^2(t) = 2(1 - \cos(2t))$$

$$\stackrel{\mathcal{F}}{\iff} X(\nu) = 2\delta(\nu) - \delta\left(\nu - \frac{1}{\pi}\right) - \delta\left(\nu + \frac{1}{\pi}\right)$$

<4 pts>

1.(b)

La discrétisation (tous les T_e) dans l'espace primal (temporel ou spatial) est équivalente à une périodisation dans l'espace des fréquences (tous les $\nu_e = 1/T_e$). $X(\nu)$ va donc se périodiser toutes les fréquences multiples de $\nu_e = 4/\pi$. Analytiquement on aura donc :

$$X_e(\nu) = X(\nu) * \text{III}_{1/T_e=4/\pi}(\nu)$$

$$X_e(\nu) = \sum_{n=-\infty}^{+\infty} \left\{ 2\delta\left(\nu + \frac{4n}{\pi}\right) - \delta\left(\nu - \frac{1-4n}{\pi}\right) - \delta\left(\nu + \frac{1-4n}{\pi}\right) \right\}$$

<3 pts>

Oui la condition de Shannon est respectée car la fréquence d'échantillonnage $\nu_e = 4/\pi > 2 \cdot (\nu_0 = 1/\pi)$ avec $\nu_0 = 1/\pi$ la fréquence maximale contenue dans le signal $x(t)$.

<2 pts>

2.(a)

La fonction $x_T(t)$ est impaire, sa TF sera donc imaginaire pure. $x_T(t)$ étant de plus une fonction réelle, sa TF sera donc à symétrie hermitienne. Sa TF sera donc imaginaire pure et impaire.

De plus, le signal étant non périodique, sa TF sera continue et $x_T(t)$ possédant deux discontinuités (à l'origine), sa TF demandera beaucoup d'harmoniques pour reconstruire cette discontinuité ce qui se traduira par un support fréquentiel assez grand (cette dernière remarque sera d'autant plus vraie lorsque T tendra vers 0, faisant apparaître une discontinuité pour laquelle la dérivée en ce point sera infinie).

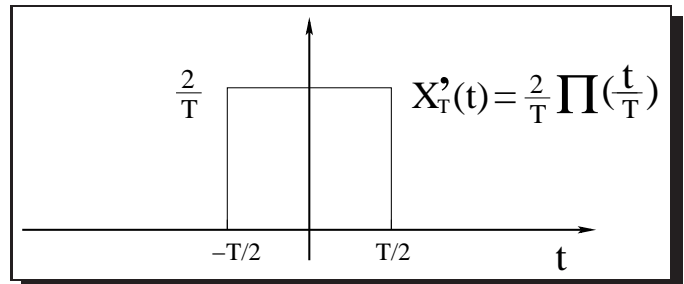
<4 pts>

2.(b)

On trouve facilement que $x'_T(t)$ s'exprime à partir de la fonction porte (appelée aussi fonction ouverture ou rectangle), i.e., :

$$x'_T(t) = \frac{2}{T} \Pi\left(\frac{t}{T}\right)$$

<4 pts>



2.(c)

Avec la correspondance $t \iff \nu$, on va se servir de la propriété de dérivation qui stipule que $\mathcal{F}[f'(t)] = (2\pi j\nu) F(\nu)$ et bien sur de la relation précédente pour écrire :

$$x'_T(t) = \frac{2}{T} \Pi\left(\frac{t}{T}\right) \stackrel{\mathcal{F}}{\rightleftharpoons} (2\pi j\nu) F(\nu) = 2 \frac{\sin \pi\nu T}{\pi\nu T}$$

qui nous permet de trouver $F(\nu) = \mathcal{F}[f(t)] = \mathcal{F}[x_T(t)]$:

$$\mathcal{F}[x_T(t)] = \frac{1}{\pi j\nu} \frac{\sin \pi\nu T}{\pi\nu T} = -j \frac{\sin(\pi\nu T)}{\pi^2\nu^2 T}$$

<4 pts>

Nota : A ce stade, on peut vérifier que ce qui avait été intuité à la question 1.(a) est vraie ; à savoir que la TF de $x_T(t)$ est continue, imaginaire pur et impair et à grand support fréquentiel.

2.(d)

Quand $\lim_{T \rightarrow 0}$, on retrouve la fonction signe, notée $\text{sgn}(t)$ et pour laquelle :

$$\text{sgn}(t) = \begin{cases} 1 & \text{si } t > 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases}$$

dont la TF devrait être $\lim_{T \rightarrow 0} \left\{ \frac{1}{\pi j\nu} \frac{\sin \pi\nu T}{\pi\nu T} \right\} = F(\nu) = \frac{1}{\pi j\nu}$

Et on n'aurait pas pu calculer cette TF à partir de la définition intégrale sur $x_T(t)$ car celle-ci n'est pas carré ou absolument sommable.

<4 pts>

2.(e)

En utilisant maintenant la propriété duale sur la relation précédente, i.e., $\mathcal{F}[\text{sgn}(t)] = 1/(\pi j\nu)$, on trouve immédiatement $\mathcal{F}^{-1}[\text{sgn}(\nu)] = -1/(\pi j t)$ et donc $\mathcal{F}[\frac{1}{t}] = \pi \text{sgn}(\nu)$.

$$\text{On a donc : } \frac{1}{t} \stackrel{\mathcal{F}}{\rightleftharpoons} \pi \text{sgn}(\nu)$$

<4 pts>

3.(a)

Avec la correspondance $f(x) \stackrel{\mathcal{F}}{\rightleftharpoons} F(\nu)$ et $g(x) \stackrel{\mathcal{F}}{\rightleftharpoons} G(\nu)$, on a :

$$\begin{aligned} \frac{\partial^n}{\partial x^n} \{f(x) * g(x)\} &\stackrel{\mathcal{F}}{\rightleftharpoons} (2\pi j\nu)^n \cdot F(\nu) \cdot G(\nu) \\ &= \left\{ (2\pi j\nu)^{n/2} \cdot F(\nu) \right\} \cdot \left\{ (2\pi j\nu)^{n/2} \cdot G(\nu) \right\} \stackrel{\mathcal{F}^{-1}}{\rightleftharpoons} \left\{ \frac{\partial^{n/2}}{\partial x^{n/2}} f(x) \right\} * \left\{ \frac{\partial^{n/2}}{\partial x^{n/2}} g(x) \right\} \end{aligned}$$

<5 pts>

3.(b)

Si on remplace $n = 2$ dans la relation de l'Eq. (1) et si on considère que $f(x)$ correspond à l'image et $g(x)$ au masque gaussien. Le filtre passe-bande (de Marr-Hildreth), qui correspond au calcul de la dérivée seconde (via son masque de convolution) sur l'image précédemment convoluée par un masque gaussien correspond bien à l'opération du membre de gauche de cette relation.

Du coup, l'opération précédente est équivalente aux deux opérations du membre de droite de cette relation, à savoir ; la calcul du gradient de l'image (par convolution, par exemple, avec le masque $\begin{bmatrix} -1 & 1 \end{bmatrix}$), puis la convolution du résultat obtenu par un masque de convolution qui serait la discrétisation de la dérivée (première) d'une Gaussienne.

Nota : De même qu'il aurait été tout aussi équivalent de convoluer l'image avec la dérivée seconde de la gaussienne.

<5 pts>

II. Opérations Ponctuelles (18 pts)

- Pour avoir une mesure de contraste qui soit proche de la mesure de contraste de brillance prise en compte par le système visuel humain et que l'on puisse aussi calculer sur des images naturelles et permettant de quantifier le rehaussement de contraste créé par un algorithme, deux auteurs proposent les deux mesures suivantes (pour une image I supposée toroidale de taille $N \times N$ et de luminance $f(x, y)$) :

$$C_1 = \sum_{x=0}^N \sum_{y=0}^N \left\{ [f(x+1, y) - f(x, y)]^2 + [f(x, y+1) - f(x, y)]^2 \right\} \quad (2)$$

$$C_2 = \frac{1}{N^2} \sum_{x=0}^N \sum_{y=0}^N C_l(x, y) \quad (3)$$

$$\text{avec } C_l(x, y) = \frac{f_{\max}(x, y) - f_{\min}(x, y)}{f_{\max}(x, y) + f_{\min}(x, y)}$$

et $f_{\max}(x, y)$ (resp. $f_{\min}(x, y)$) représentant l'intensité maximale et minimale contenu dans une petite fenêtre carrée de taille $m \times m$ centrée sur le pixel de coordonnées (x, y) .

Discuter de l'intérêt et de la pertinence de ces deux mesures de contraste comparativement à la mesure de contraste habituellement prise en traitement d'images utilisant l'écart type des variations de gris dans l'image.

<9 pts>

```

void Compute_histo(float** img, int lgth, int wdth, float* hist)
{ int i, j;

  for(i=0; i<GREY_LEVEL; i++) hist[i]=0.0;

  for(i=0; i<lgth; i++) for(j=0; j<wdth; j++)
  if ((img[i][j]>0)&&(img[i][j]<GREY_LEVEL)) hist[(int)(img[i][j]*0.5)]++;

  for(i=0; i<GREY_LEVEL; i++) hist[i]=((float)(wdth*lgth)); }

//-----
// Specif: img_in possédant le même histo que img_out
void Specif(float** img_in, int lgth_in, int wdth_in, float** img_out, int lgth_out, int wdth_out)
{ int i, j;
  float Hist_img1[GREY_LEVEL+1], Hist_img2[GREY_LEVEL+1];
  float Vect_img1[GREY_LEVEL+1], Vect_img2[GREY_LEVEL+1], Vect_img3[GREY_LEVEL+1];

  //Initialisation
  for(i=0; i<GREY_LEVEL; i++)
  { Hist_img1[i]=Hist_img2[i]=0.0;
    Vect_img1[i]=Vect_img2[i]=Vect_img3[i]=0.0; }

  //Calcule histogramme_in
  Compute_histo(img_in, lgth_in, wdth_in, Hist_img1);

  //Calcule T::fonctRepartition[img_in]>Vect_img1
  for(i=0; i<GREY_LEVEL; i++)
  { if (i>0) Vect_img1[i]=Vect_img1[i-1]+Hist_img1[i];
    else Vect_img1[i]=Vect_img1[i]; }
  for(i=0; i<GREY_LEVEL; i++) Vect_img1[i]=((int)((Vect_img1[i]*255)+0.5));

  //Egalise histogramme_in
  for(i=0; i<lgth_in; i++) for(j=0; j<wdth_in; j++)
  img_in[i][j]=((int)>Vect_img1[(int)(img_in[i][j])]);

  //-----
  //Calcule histogramme_out
  Compute_histo(img_out, lgth_out, wdth_out, Hist_img2);

  //Calcule G::fonctRepartition[img_out]>Vect_img2
  for(i=0; i<GREY_LEVEL; i++)
  { if (i>0) Vect_img2[i]=Vect_img2[i-1]+Hist_img2[i];
    else Vect_img2[i]=Vect_img2[i]; }
  for(i=0; i<GREY_LEVEL; i++) Vect_img2[i]=((int)((Vect_img2[i]*255)+0.5));

  ...
  ?
  ...
}

```

FIG. 1 – Code à compléter

- Voici le début du code en C (cf. Fig. 1) d'une fonction en langage C qui réalise une spécification d'histogramme de l'image img_{in} en niveaux de gris sur l'image img_{out} . Compléter les quatre ou cinq lignes qui lui manquent pour rendre le programme fonctionnel.

<9 pts>

Réponse

1

Remarquons avant tout que ces deux mesures sont locales et moyennées globalement et non globales comme le calcul des mesures de contrastes vues en cours. Cette propriété permettra d'obtenir une plus grande fiabilité de la mesure de contraste.

Considérons les deux paires d'images tests que l'on avait pris en cours, à savoir le petit carré de niveaux de gris 0.3 centré sur un fond de niveaux de gris 0.1 comparativement à l'image du petit carré de niveaux de gris 0.7 centré sur un fond de niveaux de gris 0.5 (avec la convention 0-noire et 1-blanc) et les deux images de Mona, l'une normale et l'autre avec une variation des niveaux de gris moins importante (moindre contraste) mais possédant néanmoins un (seul) pixel noir et un (seul) pixel blanc.

Soit aussi C_3 , la mesure utilisant l'écart type des variations des niveaux de gris dans l'image vue en cours et habituellement utilisée comme mesure de contraste.

- La première est en fait la norme l^2 du gradient calculé globalement sur toute l'image et les mesures de contraste C_1 et les conclusions que l'on obtiendra sur ces deux paires d'images seront les mêmes que celle donnée par la mesure utilisant l'écart type des variations des niveaux de gris dans l'image. Pour la première paire d'images, elle donnerait le même contraste alors que le système visuel humain (SVH) nous laisse penser le contraire. De plus, notons aussi que cette mesure (du gradient) aura du mal à distinguer une image effectivement contrastée d'une image en fait bruitée.

- La deuxième formule de contraste est la mesure locale moyennée (sur toute l'image) de la (deuxième) formule vue en cours. Elle permettrait, contrairement à la mesure globale vue en cours, de trouver une mesure de contraste adéquate pour la deuxième paire d'images (malgré le fait que l'image de Mona pourrait posséder n (seul) pixel noir et un (seul) pixel blanc. De plus cette mesure permettrait, pour la première paire d'images, d'avoir deux mesures de contraste qui seraient différentes et en accord avec ce que pense le système visuel humain. Par contre, cette mesure serait différente pour différentes valeurs du paramètre m que l'on devrait préalablement estimer et demanderait plus de coût calculatoire.

Une autre caractéristique intéressante de C_2 ; prenons le cas deux images de Mona, l'original et celle dont la moyenne de l'image est translaté de 0.1 ou 0.2 vers les zones sombres (avec la convention 0-noire et 1-blanc). Ces deux images ont le même contraste (donné par C_3 ou par l'écart type). Or cette mesure C_2 , à cause du terme de division $f_{\max}(x, y) + f_{\min}(x, y)$ donnera un contraste plus grand pour la plus sombre des deux images. Cette propriété est similaire au SVH (à cause du phénomène de saturation) où notre perception du contraste baisse pour les intensité élevé bien que dans le cas du SVH cela se fait logarithmiquement et non pas linéairement.

<9 pts>

2

Il reste simplement le calcul de G^{-1} puis le codage de la spécificité d'histogramme utilisant la fonction G^{-1} .

```
...
//Calcule G^-1 (avec G=Vect_Img2)>Vect_Img3
for(i=0;i<=GREY_LEVEL;i++)
  { Vect_Img3[i]=i;
    for(j=GREY_LEVEL;j>0;j--) if (Vect_Img2[j]>i) Vect_Img3[i]=j; }

//Specif. histo[img_in]>histo[img_out]
for(i=0;i<lgth_in;i++) for(j=0;j<wdth_in;j++)
  img_in[i][j]=(int)Vect_Img3[(int)(img_in[i][j])];
}
```

<9 pts>

III. Filtrage (35 pts)

Filtrage Spatial

1. Soit le masque de convolution 1×3 suivant : $\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$
 - (a) Rappeler pourquoi ce masque de convolution, que l'on appelle souvent masque Gaussien, n'est en fait qu'une approximation de la discrétisation de la Gaussienne.
<4 pts>
 - (b) Donner une estimation ou un intervalle d'estimation, exprimée en nombre de pixels, de l'écart type de la Gaussienne que ce masque est censé approximer.
<6 pts>

2. Soit le filtre précédent en 2D : $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

On aimerait implémenter ce masque de convolution non pas logiciellement mais en utilisant les capacités d'un DSP (processeur de traitement des signaux) pour optimiser la vitesse de ce traitement. On rappelle qu'un DSP implémente (en hardware) que des convolutions de filtres moyenneurs (*average filter*) de n'importe quelle taille possible (pour des masques moyenneurs pas forcément carrés). Indiquer la décomposition de ce filtre puis la propriété utilisée pour son implémentation sur DSP.

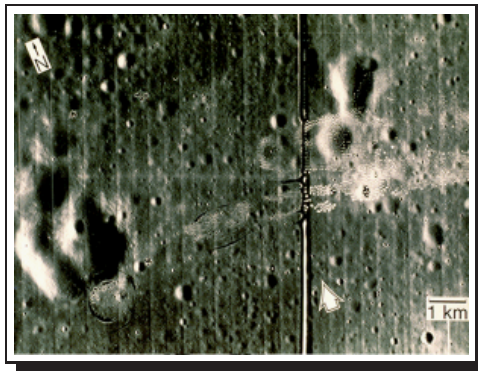
<6 pts>

3. Soit le filtre passe-bas adaptatif (vue en cours) dans lequel $f(x, y)$ correspond à une image bruitée par du bruit Gaussien, $g(x, y)$ l'image débruitée par ce filtre et PB, un filtre ou masque de convolution passe-bas du style masque Gaussien ou moyeneur et :

$$g(x, y) = \begin{cases} \text{filtre PB}[f(x, y)] & \text{si } |\text{filtrePB}[f(x, y)] - f(x, y)| < \text{seuil} \\ f(x, y) & \text{sinon} \end{cases}$$

- (a) Si on connaissait la variance du bruit Gaussien entachant l'image $f(x, y)$, quelle serait la valeur du seuil idéale?
<3 pts>
- (b) Si on ne connaissait pas cette variance, comment pourrait-on l'estimer sur l'image bruitée?
<4 pts>
- (c) Est-ce que cette technique serait intéressante si l'image était entachée par du bruit poivre et sel? Justifier bien votre réponse.
<4 pts>

Filtrage Fréquentiel



4. En digitalisant une vieille carte de la surface de la lune, on a aussi numérisé la ligne blanche verticale située à droite de cette carte et qui devait correspondre au pli de l'image de cette photo analogique

prise au début des années 60. On aimerait enlever cette dégradation en utilisant la Transformée de Fourier (TF).

Indiquer d'une part comment cette dégradation influencera le module de la TF de cette image et quelle pourrait être l'opération fréquentielle permettant d'atténuer cette dégradation.

<8 pts>

Réponse

1.(a)

La Gaussienne est une fonction continue à support infini; elle ne peut être discrétisée sur un support fini qu'avec une certaine approximation. De plus son amplitude, à différent intervalles (dans le cas discret), est approximée sous forme de fractions d'entiers.

Connaissant l'équation de la gaussienne : $\propto \exp(-\frac{x^2}{2\sigma^2})$, l'abscisse x permettant d'obtenir une amplitude à mi hauteur serait $x \approx 1.17\sigma$ où σ serait l'écart type de la gaussienne (et une estimation possible de l'écart type de la gaussienne que ce masque de convolution approxime, comme nous le demande la question suivante, en nombre de pixels, serait donc de $1/1.17 \approx 0.85$ pixels). Cependant pour $x = 1.17 \times 2 = 2.34$, l'amplitude de la gaussienne est $\exp(-(2.34^2/2)) \approx 0.064$ et non pas 0.0 et pour $x = 1.17 \times 3 = 3.51$, l'amplitude de la gaussienne est $\exp(-(3.51^2/2)) \approx 0.002$ et non pas 0.0.

Sans facteur de normalisation, une (meilleure) approximation d'une gaussienne (à 1% près) pour un écart type de $\sigma = 0.85$ pixels serait donc le filtre de longueur 5 ou 7 colonnes suivant :

$$\begin{array}{|c|c|c|c|c|} \hline 0.064 & 1 & 2 & 1 & 0.064 \\ \hline \end{array} \quad \text{ou} \quad \begin{array}{|c|c|c|c|c|c|} \hline 0.002 & 0.064 & 1 & 2 & 1 & 0.064 & 0.002 \\ \hline \end{array}$$

Nota : Dans le cours, il est dit qu'il faudrait pour approximer la gaussienne (à au moins 1% près), un masque de longueur (en pixels) $6\sigma + 1$, ici, cela donnerait une longueur de $(6 \times 0.85) + 1 \approx 6$.

<4 pts>

1.(b)

En fait il existe deux façons d'estimer σ à partir de ce masque de convolution approxinant une gaussienne. Ces deux façons nous donneront un interval d'estimation pour ce paramètre.

- La première façon est de dire que l'abscisse x qui permet d'obtenir une amplitude à mi-hauteur de la gaussienne ($\propto \exp(-\frac{x^2}{2\sigma^2})$) est $x \approx 1.17\sigma$. Une estimation possible de σ , en nombre de pixels, est donc de $1/1.17 \approx 0.85$ pixels.

- Une autre façon d'estimer σ à partir de ce masque qui approxime une gaussienne est de dire que dans le cas de ce masque, pour deux pixels éloignés du centre de ce masque, la valeur du masque est 0 et que cela doit correspondre à 3σ . σ est donc supérieur à $2/3 \approx 0.\overline{6}$ pixels.

$$\text{On obtient donc :} \quad 2/3 \approx 0.\overline{6} \text{ pixels} < \sigma < 0.85 \text{ pixels}$$

Nota : Une autre démonstration est de dire que l'écart type de la Gaussienne est relié à sa largeur mi-hauteur (FWHM) par la relation :

$$\sigma = \frac{FWHM}{2\sqrt{2\ln(2)}}$$

et comme dans notre cas $FWHM = 2$, on trouve $\sigma \approx 0.85$.

<6 pts>

2

On a : $M = M_1 + M_2 + M_3 + M_4$ avec :

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{16} \left\{ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \right\}$$

et on utilisera la propriété de distributivité/associativité du produit de convolution pour dire que (si I est l'image) :

$$I * M = (I * M_1) + (I * M_2) + (I * M_3) + (I * M_4)$$

Nota -1- : Une autre démonstration (beaucoup plus courte) est de dire que le masque Gaussien est séparable en un filtre $(1/4) \{ \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T \}$ où l'exposant T désigne la transposée et que l'on peut décomposer facilement ces deux filtres en deux filtres moyennés de taille 3 et 1 colonne(s), en notant simplement que : $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix}$.

Nota -2- : Avec ce que l'on a dit en Nota -1-, on pourrait décomposer $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix}$ et donc re-crée le masque gaussien, avec un DSP, en filtrant deux fois horizontalement avec $\begin{bmatrix} 1 & 1 \end{bmatrix}$ et deux fois verticalement avec $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$ (puis diviser le résultat par la constante 16).

Nota -3- : On pourrait aussi décomposer le masque M en deux filtres 2×2 moyennés selon :

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

<6 pts>

3.(a)

3 ou 4 σ . Cette erreur correspondra à une borne supérieure de la différence en valeur absolue de l'image filtrée PB (supposée quasi-débruitée) avec l'image bruitée (par du bruit Gaussien d'écart type *sigma*). L'erreur faite entre les transitions filtrées PB et les transitions bruitées serait beaucoup plus grande.

<3 pts>

3.(b)

Une méthode possible consisterait, pour chaque pixel, à estimer dans une petite fenêtre W (par exemple de dimension 5×5) et centrée sur ce pixel, la variance empirique. La variance minimale obtenue sur l'ensemble de ces estimations serait une borne supérieure et une assez bonne estimation de la variance de ce bruit. En effet l'image comporte certainement quelques zones homogènes et lorsque W se fixera sur l'une d'entre elles, la variance obtenue sur cette zone sera une assez bonne estimation de la variance du bruit.

<4 pts>

3.(c)

Non car d'une part le PB serait inefficace sur ce bruit impulsionnel quel que soit la valeur du seuil. En effet, comme le bruit "poivre et sel" crée des pixels dont les niveaux de gris sont proches de "0" et "255", le terme $|\text{filtrePB}[f(x,y)] - f(x,y)|$ sera très grand partout et pas seulement aux niveaux des transitions existant dans l'image. Du coup, l'algorithme n'est plus adaptatif.

<4 pts>

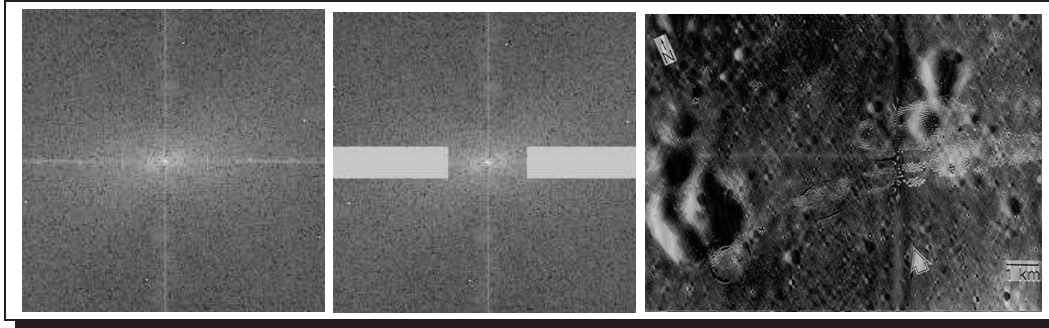
4

À cause de cette ligne blanche verticale faisant apparaître une discontinuité verticale, on s'attend à ce que le module de la TF de cette image laisse apparaître une ligne blanche horizontale centrée dans son spectre qui correspondrait en fait à de forts coefficients spectraux nécessaires pour reconstruire l'image avec sa dégradation.

L'opération fréquentielle qui éliminerait la partie haute fréquence de cette ligne blanche horizontale dans le spectre de l'image serait donc l'opération fréquentielle appropriée permettant d'atténuer cette dégradation.

<8 pts>

Nota : En appliquant la stratégie mentionnée, on obtiendrait :

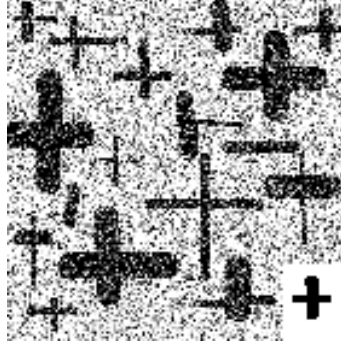


VI. Misc. (32 pts)

1. Soit une image carrée de taille $N \times N$ pixels. Quelle serait la complexité algorithmique d'une procédure de détection d'une courbe paramétrique dépendante de p paramètres par une transformée de Hough?
<5 pts>
2. Supposons que vous deviez estimer la PSF (ou fonction de floue) d'une voiture en mouvement (prise par une caméra électronique) afin de pouvoir utiliser cette PSF pour déconvoluer ensuite sa plaque d'immatriculation (afin de pouvoir lire le numéro qui y figure). Dans ce cas de figure, citer tous les avantages (en justifiant bien votre réponse) de posséder non pas une image floue mais une séquence de n images données par une caméra fixe.
<5 pts>



3. Discuter brièvement quelles sont les ressemblances/différences avantages/inconvénients de la méthode de restauration de Figueiredo et Nowak (FN) vue en TP et celle de Tichonov et Miller (TM) vue en cours.
<5 pts>



4. Proposer deux méthodes pour filtrer l'image (binaire) ci-dessus afin de supprimer le bruit et décrire une démarche permettant d'extraire automatiquement toutes les croix pouvant contenir la croix du coin inférieur droit.
<5 pts>
5. Donner le code en langage C de la fonction `REQUANTIFIEDCOLORHISTOGRAM(FLOAT*** IMG,FLOAT* VCTFEATURES,INT NBBIN,INT WDTHLGTH)` qui estime l'histogramme couleur requantifié normalisé sur `NbBin` bins rouges, `NbBin` bins verts et `NbBin` bins bleus de l'image couleur `img` de même longueur et largeur : `WdthLgth` et stocke les valeurs de cet histogramme dans le vecteur de flottant `VctFeatures` de dimension `NbBin × NbBin × NbBin`.
<7 pts>
6. Pour caractériser une texture, On décide de prendre les 4^3 valeurs de son histogramme requantifié couleur normalisé comme vecteur de caractéristiques. Discuter de l'invariance de ses paramètres.
<5 pts>
7. Quel est l'avantage de l'espace couleur LAB que j'ai énoncé en cours.
<4 pts>

Réponse

1

On doit d'abord parcourir l'image de contour, soit demander une complexité $O(N^2)$ et pour chacun de ces points de contour parcourir ensuite une matrice d'accumulation de même taille mais de dimension p (correspondant au p paramètres discrétisés de la fonction paramétriques), ce qui se fait avec une complexité de $O(N^p)$. En tout donc on a une complexité de $O(N^{p+2})$.

<5 pts>

2

Je vois deux avantages :

- Le premier avantage est de donner la direction de la PSF ou fonction de floue. On sait, puisqu'il s'agit d'un flou de mouvement, qu'il s'agit d'un vecteur de translation pour lequel il faut donc estimer son amplitude (en nombre de pixels) mais aussi sa direction. La direction sera donnée facilement en voyant comment un point (par exemple le point le plus à gauche et en bas de la plaque d'immatriculation) se déplace d'une image à l'autre. Par contre, on ne peut estimer l'amplitude du vecteur car le temps d'ouverture du diaphragme de la caméra peut être plus petit que le temps existant entre l'acquisition de deux images par la caméra.

- Le deuxième avantage est que l'on peut moyenner l'estimation de la PSF obtenue (sur un petit nombre d'images, supposant ainsi que le vecteur de mouvement ne bouge pas trop durant un laps de temps bref) ou encore moyenner le résultat de la déconvolution obtenu sur l'image I_t avec celui obtenu sur l'image I_{t+1} , I_{t+2} , etc.

<5 pts>

3

Toutes les deux sont des méthodes itératives. La fonction de vraisemblance est la même dans les deux cas et conduit à une itération de l'algorithme de Landweber. La régularisation est différente. Contrairement à TM ou la régularisation s'exprime par un terme d'énergie qui défavorise les solutions ayant des hautes fréquences dans le domaine de Fourier (*via* la norme L2 du laplacien de l'estimation), la régularisation de l'algorithme de FN se fait en alternant une itération de Landweber avec un *denoising* dans le domaine des ondelettes. L'avantage de faire un débruitage dans le domaine des ondelettes (plutôt que dans Fourier) est de mieux conserver les contours de l'image restaurée et de ne pas introduire de *ringing* dans la solution (i.e., l'image déconvoluée) car la suppression des fréquences se fait localement

<5 pts>

4

1. Filtrage médian ou ouverture suivie de fermeture.
2. Image initiale moins l' image érodée par l'élément.

<5 pts>

5

```
void RequantifiedColorHistogram(float*** img,
float* UctFeatures, int NbBin, int WdthLgth)
{ int i, j, k;
  int r, v, b;
  int idx;
  int WdthBin=1+255/NbBin;

  //Init
  for(k=0; k<NbBin*NbBin*NbBin; k++) UctFeatures[k]=0.0;

  //Histo
  for(i=0; i<WdthLgth; i++) for(j=0; j<WdthLgth; j++)
  {
    r=(int)(img[0][i][j]/WdthBin);
    v=(int)(img[1][i][j]/WdthBin);
    b=(int)(img[2][i][j]/WdthBin);
    UctFeatures[(r*NbBin+NbBin)*v+NbBin+b]++;
  }

  //Normalization
  for(k=0; k<NbBin*NbBin*NbBin; k++)
  UctFeatures[k]/=(NbBin*NbBin*NbBin);
}
```

<7 pts>

6

Totalement invariant par rotation, très invariant relativement au bruit qui pourrait s'ajouter sur une image naturelle, assez robuste à un facteur d'échelle pas trop grand, assez peu robuste aux ombres et changement de luminosité qui pourrait affecter la texture.

<5 pts>

7

A cause des relations non-linéaires entre L, A et B (qui imite quelque peu le SVH), la distance existante entre deux couleurs représentera bien la différence, perçue par le système visuel humain (SVH), existante entre ces deux couleurs.

Dit d'une autre façon, l'espace couleur LAB permet de mieux représenter les écarts entre les couleurs que perçoivent les yeux humains.

<4 pts>