DIRO
IFT 3205

TRAVAIL PRATIQUE

Intercorrélation - Interspectre (1)

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle.
http : //www.iro.umontreal.ca/∼mignotte/ift3205
e-mail : mignotte@iro.umontreal.ca

1 Introduction

L'objectif de ce TP est d'étudier les propriétés de la fonction d'intercorrélation (et sa FFT) et de voir son utilitée dans le cadre de l'analyse des signaux et du traitement du signal. La fonction d'intercorrélation qui est un outil important en traitement du signal car elle permet principalement de mesurer la ressemblance entre deux signaux f et g. En supposant que f et g sont des signaux réels unidimensionels dans le domaine spatial, la fonction d'intercorrelation C_{fg} est défini par :

$$C_{fg}(\tau) \triangleq \int f(x)\, g(x-\tau)\, dx$$

C_{fg} quantifie dans quelle mesure on peut écrire $f(x) = \alpha g(x-\tau) + \beta$ et dispose des propriétés suivantes :

- $|C_{fg}(\tau)|^2 \leq C_{ff}(0)\, C_{gg}(0)$, $\forall \tau$ -inégalité de SCHWARZ-
- $C_{fg}(\tau) = C^{\star}_{gf}(-\tau)$ -symétrie hermitienne-
- $0 \leq |C_{fg}(\tau)| \leq 1$ -après avoir normalisé cette fonction en la divisant par sa valeur maximale $C^{\max}_{fg}$-

Remarquer que la convolution est une intercorrelation dans lesquel un des signaux est renversée dans l'espace. Inversement, l'intercorrélation peut donc se noter à partir d'une convolution de la façon suivante

$$C_{fg}(\tau) \triangleq f(\tau) * g(-\tau) \quad \overset{\mathcal{F}}{\rightleftharpoons} \quad F(\nu)\, G^{\star}(\nu) \triangleq \mathcal{C}_{fg}(\nu)$$

avec F et G, les TF de f et g et en se souvenant que, avec la correspondance $\tau \overset{\mathcal{F}}{\rightleftharpoons} \nu$, on a $f(-\tau) \overset{\mathcal{F}}{\rightleftharpoons} F^{\star}(\nu)$. $\mathcal{C}_{fg}(\nu)$, la TF de la fonction d'intercorrélation, désigne la densité interspectrale d'énergie et caractérise la contribution énergétique des différentes contributions fréquentielles.

2 Recalage d'Image ou de Signaux

La fonction d'intercorrélation et la TF sont deux outils puissants utilisés en traitement du signal et des images qui permettent de faire, entre autres, le recalage c'est-à-dire la mise en correspondance de signaux ou d'images, ceci afin de pouvoir comparer ou combiner leurs informations respectives. Cette mise en correspondance se fait par la recherche d'une transformation géométrique (e.g., translation, facteur d'échelle, rotation, etc.) permettant de passer d'une image (ou un signal 1D) à une autre. Cette technique a de nombreuses applications allant de l'imagerie médicale afin par exemple de fusionner plusieurs modalités d'imagerie, au traitement de vidéos comme le suivi de mouvement, la compression, le recalage de profils sismiques, ou encore la création de mosaïques d'images (panoramas).

Considérons deux images $f(x,y)$ et $g(x,y)$ à recaler. Si on considère qu'il existe un décalage, due à une translation (de vecteur (x_0, y_0)) entre ces deux images, alors on peut écrire grâce à une propriété de la TF

$$g(x,y) = f(x - x_0, y - y_0) \quad \overset{\mathcal{F}}{\rightleftharpoons} \quad G(u,\nu) = F(u,\nu) \exp\big[-2\pi j(ux_0 + \nu y_0)\big]$$

avec la correspondance $x, y \overset{\mathcal{F}}{\rightleftharpoons} u, \nu$. Ceci signifie que les images f et g ont le même module de Fourier, i.e., $|G(u,\nu)| = |F(u,\nu)|$ tandis que la différence de phase est directement reliée au déplacement spatial existant entre ces deux images. De plus, comme $F(u,\nu) = |F(u,\nu)| \exp[-j\phi_f(u,\nu)]$, on trouve

$$\frac{\overbrace{G(u,\nu)\, F^{\star}(u,\nu)}^{\mathcal{C}_{gf}(u,\nu)}}{|F(u,\nu)|^2} \quad = \quad \exp[-2\pi j(ux_0 + \nu y_0)]$$

$$\mathcal{F}^{-1}\left[\frac{\mathcal{C}_{gf}(u,\nu)}{|F(u,\nu)|^2}\right] = \delta(x - x_0, y - y_0) \tag{1}$$

Si on considère maintenant qu'il existe, en plus de la translation, une rotation entre $f(x,y)$ et $g(x,y)$, c'est-à-dire une transformation totale (translation + rotation) permettant d'écrire entre les images f et g, la relation $g(x,y) = f(x\cos\theta_0 + y\sin\theta_0 - x_0, -x\sin\theta_0 + y\cos\theta_0 - y_0)$ alors, selon une autre propriété de la TF, on montre que, pour des valeurs raisonablement pas trop grandes de (x_0, y_0), $|F(u,\nu)|$ est une réplique pivotée de $|G(u,\nu)|$ suivant l'angle θ_0 et on a donc

$$|G(u,\nu)| = |F(u\cos\theta_0 + \nu\sin\theta_0, -u\sin\theta_0 + \nu\cos\theta_0)|$$

Donc, si on résume, une fois calculé $|F(u,\nu)|$ et $|G(u,\nu)|$, une mesure de corrélation entre $|F(u,\nu)|$ et $|G(u,\nu)|$ pour différentes valeurs d'angle de rotation appliquée à $|G(u,\nu)|$ permet de trouver θ_0 puis d'utiliser ce paramètre pour qu'il n'y ait plus qu'un décalage (i.e., une translation) entre f et g. Le calcul ensuite de la TF inverse de la densité interspectrale entre ces deux fonctions permet de trouver (x_0, y_0) et par translation inverse de recaler g sur f. Dans la suite on va considérer que l'image UdM_1.pgm est l'image f et que l'image UdM_2.pgm est l'image g. On vous demande donc dans ce TP de

1. Calculer $|F(u,\nu)|$ et $|G(u,\nu)|$. Visualiser les deux spectres en utilisant ce qui est nécessaire (cf. TP précédent) pour visuellement apprécier leur différence.

Fig. 1 – *De gauche à droite. Les deux images f (UdM_1.pgm) et g (UdM_2.pgm) considérées. Image g recalée en rotation et image de l'interspectre de ces deux dernières images ; on voit le $\delta(x - x_0, y - y_0)$ (point blanc) de l'Équation (1).*

2. Programmer une fonction permettant de faire pivoter une image selon un angle θ_0 donné et utilisant l'interpolation bilinéaire[1] (si vous avez des difficultés avec l'interpolation bilinéaire, utiliser l'interpolation au plus proche voisin[1]). Expérimenter votre fonction sur l'image lena avec un angle $\theta_0 = 22.5$ degré.
3. Utiliser la fonction précédente pour faire pivoter $|G(u,\nu)|$ pour différentes valeurs discrètes de θ_0 (e.g., entre $-\pi/16$ et $\pi/16$ par pas de 0.005 radian) et pour calculer une mesure d'erreur de ressemblance du type $\epsilon = \sum_{u,v} ||G_{\theta_0}(u,\nu)| - |F(u,\nu)||$, où $|G_{\theta_0}|$ désigne le module de la TF de G pivoté de θ_0 et faîtes un programme qui permette de retenir (i.e., estimer) automatiquement la valeur θ_0 assurant la ressemblance maximale, c'est-à- dire une mesure d'erreur ϵ (de ressemblance) minimale.

4. Faire l'opération de rotation inverse (par interpolation) pour s'assurer que l'image g n'a plus qu'une différence de décalage (cf. image 1.(c)).
5. Calculer l'estimation du vecteur de translation entre l'image f et l'image précédente en utilisant l'Équation (1) (cf. image 1.(d)).
6. faîtes un programme qui permette de Recaler ces deux images pour obtenir l'image panoramique de la Fig. 2.

Fig. 2 – *Image recalée en rotation et translation*

Annexe

On rappelle dans cette annexe, quelques notions utiles pour faire une opération réussie de rotation d'image avec une interpolation bilinéaire.

1. **Rotation** : À la place de faire une TRANSFORMATION SPATIALE DIRECTE qui consisterait à prendre chaque pixel de niveau de gris en (x, y) dans l'image originale (image 3a) puis à placer celui-ci en (x', y') dans l'image transformée (cf. Fig. 3b) suivant la relation $x' = x \cos\theta - y \sin\theta$ et $y' = x \sin\theta + y \cos\theta$, on préfère faire l'opération de TRANSFORMATION SPATIALE INVERSE qui consiste à remplir (selon un balayage lexicographique) chaque pixel de l'image transformée (cf. Fig. 3b) et utiliser l'opération de rotation inverse (en prenant $x = x' \cos\theta + y' \sin\theta$ et $y = -x' \sin\theta + y' \cos\theta$). Cette dernière solution est préférée car au moins on est sur que tous les pixels de l'image finale seront considérés.

(**a**) (**b**)

Fig. 3 – (**a**) *Image originale* Lenna. (**b**) *Image transformée (image précédente tournée d'un angle de 45 degré en utilisant une interpolation bilinéaire).*

[1]Prendre l'interpolation au plus proche voisin si vous n'y arrivez pas.

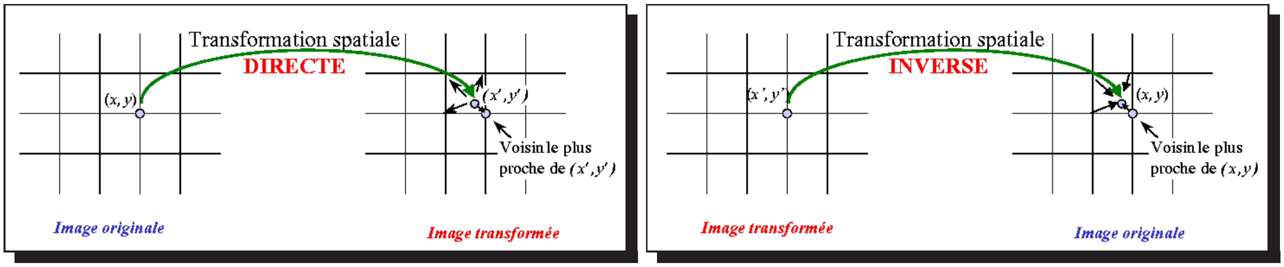

FIG. 4 –

2. **Interpolation Bilinéaire** : Lorsqu'on réalise la TRANSFORMATION SPATIALE INVERSE et que, pour chaque pixel de l'image transformée, on doit "piocher" le bon niveaux de gris dans l'image originale, il arrive que l'on se retrouve à vouloir prendre un pixel, dans l'image originale, avec une coordonnée (x, y) non entière (par exemple du style $(x = 0.3, y = 0.8)$). Dans ce cas, la solution la plus simple est de prendre le voisin le plus proche $((x = 0, y = 1.0))$ et dans ce cas, on réalise une interpolation au plus proche voisin. Une autre solution est de faire une interpolation bilinÉaire.

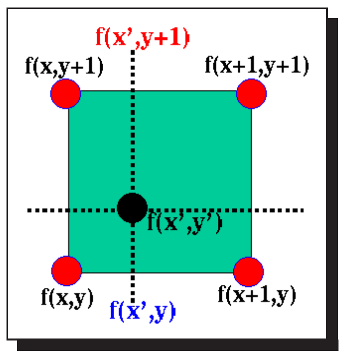

FIG. 5 –

Dans l'exemple de la figure ci-dessus, le pixel de coordonné non entière x', y' est calculé comme une somme pondéré des 4 pixels de coordonné entière qui l'encadre. On fait une interpolation bilinéaire en utilisant la formule suivante :

$$\begin{aligned} f(x', y) &= f(x, y) + (x' - x)[f(x+1, y) - f(x, y)] \\ f(x', y+1) &= f(x, y+1) + (x' - x)[f(x+1, y+1) - f(x, y+1)] \\ f(x', y') &= f(x', y) + (y' - y)[f(x', y+1) - f(x', y) \end{aligned}$$

[1]

Remise & Rapport

Vous devez rendre physiquement au démonstrateur le rapport et électroniquement le(s) programme(s) fait en C avant la date de remise spécifiée dans le fichier *barême* situé sur la page web du cours. Pour la remise électronique, utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP<Numéro du Tp>. N'oubliez pas d'inscrire vos noms, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *TpIFT3205– <Numéro du Tp>-<Numéro de la question>.c*. Les programmes devront se compiler et s'executer sur Linux tel qu'indiqué dans le barême.

Le rapport (si celui-ci est demandé dans le fichier barême) doit être concis, clair et devra contenir (en plus des éléments et des discussions que vous jugerez importants), la description brève du problème et de la technique utilisée (lorsque cela est nécesaire ou demandé), les réponses aux questions posées, les images résultats et les commentaires que l'on vous demande et que vous jugerez utile.