

IMPLEMENTATION DIGITALE DE LA TRANSFORMÉE DE FOURIER D'ORDRE FRACTIONNAIRE

E. Cousin* and Y. Torres

e-mail: ytorres@uis.edu.co

Grupo de Óptica y Tratamiento de Señales - Escuela de Física
Universidad Industrial de Santander
Bucaramanga, Colombia

*École Supérieure de Chimie – Physique - Electronique de Lyon
Villeurbanne Cedex, France

RESUMÉ

Dans cet travail on présent les résultats obtenues pour l'implémentation digitale de la transformée de Fourier d'ordre fractionnaire réel en une et deux dimensions. On fait une comparaison parmi les algorithmes disponibles.

ABSTRACT

This work presents the results for the digital implementation of the fractional Fourier transform of real order. The fractional Fourier transform is calculated for one or two dimensional signals. A comparison between existing algorithms is carried out.

1. INTRODUCTION

La transformée de Fourier Fractionnaire est un outil mathématique introduit par V. Namias^[1,2] qui l'utilise à la résolution d'équation différentielle que l'on rencontre en mécanique quantique. Néanmoins, c'est dans le domaine de l'optique, puis du traitement du signal que cette transformée a suscité le plus large intérêt. Nous nous sommes justement intéressés à l'aspect digital de ce nouvel opérateur, avec pour objectif principal la réalisation d'un logiciel permettant sa simulation et son utilisation numérique.

Nous nous concentrons sur l'aspect algorithmique de notre étude; nous nous sommes efforcés de dégager les travaux les plus intéressants de ce domaine en les analysant et en les comparant. Par ailleurs, nous décrivons leur domaine d'application avec de nouvelles perspectives d'utilisation.

Nous présentons ensuite le logiciel proprement dit, quelles sont les fonctions réalisées, l'interface associée à chacune d'elle, et expliquons leur utilisation.

2. PRÉLIMINAIRE MATHÉMATIQUE

2.1. Transformée de Fourier fractionnaire

La transformée de Fourier fractionnaire (TFf) est un opérateur à un paramètre $\alpha \in \mathbb{C}$, que l'on notera $F_\alpha = F^{(p)}$ avec $\alpha = p\pi/2$, le nombre α (ou p) se nomme ordre de la transformée. Historiquement, celle-ci a été défini initialement à partir des fonctions propres de F , la transformée de Fourier standard.

En pratique, il est plus commode de considérer cette opérateur sous sa forme intégrale; pour $0 < |p| < 2$, on le définit de la manière suivante:

$$F_\alpha[f](x) = \frac{\exp\left[i\left(\frac{\pi\tilde{\alpha}}{4} - \frac{\alpha}{2}\right)\right]}{\sqrt{2\pi|\sin\alpha|}} \exp\left(-\frac{i}{2}x^2 \cot\alpha\right) \int_{-\infty}^{\infty} \exp\left(-\frac{i}{2}x'^2 \cot\alpha\right) \exp\left(\frac{ixx'}{\sin\alpha}\right) f(x') dx' \quad (1)$$

avec $\tilde{\alpha} = \text{sgn}(\sin\alpha)$.

$$\text{pour } p=0, \quad F^{(0)}[f](x) = \int_{-\infty}^{\infty} \delta(x-x') f(x') dx' = f(x) \quad (2)$$

$$\text{pour } p=2, \quad F^{(2)}[f](x) = \int_{-\infty}^{\infty} \delta(x+x') f(x') dx' = f(-x)$$

Différentes notations seront employées: $F_\alpha[f](x) = F^{(p)}[f](x) = f_p(x)$. F_0 est donc l'opérateur identité; F_1 est la transformée de Fourier conventionnelle, et F_2 est l'opérateur d'inversion (FoF). De plus, $F^{(p)}$ est périodique de période 4, c'est-à-dire que $F^{(p)} = F^{(p+4)}$.

Les transformations de Fourier fractionnaires se composent selon la loi: $F_\alpha \circ F_\beta = F_{\alpha+\beta}$ et l'on déduit immédiatement que la transformée inverse de F_α est $F_{-\alpha}$. Ainsi, l'opérateur F_α définit un sous groupe à un paramètre du groupe unitaire de $L_2(\mathbb{R})$.

L'ensemble des propriétés précédentes justifient alors l'appellation de transformée de Fourier fractionnaire, que l'on considère comme une "puissance" de F . Par exemple, on a $F^{(1/2)} \circ F^{(1/2)} = F$ et $F^{(1/2)}$ pourrait se nommer "racine carrée" de F . Néanmoins, ceci n'est pas très rigoureux dans la mesure où il n'y a pas unicité d'un tel opérateur.

La transformée de Fourier fractionnaire reste définie sur les mêmes espaces fonctionnels que la transformée de Fourier; elle s'étend aux distributions tempérées de la façon suivante:

Si T est une distribution de $S'(\mathbb{R})$, alors:

$$\langle F_\alpha[T], \varphi \rangle = \langle T, F_\alpha[\varphi] \rangle \quad \varphi \in S(\mathbb{R})$$

Par ailleurs, On montre que F_α tend vers F_β lorsque α tend vers β . Ainsi, cet opérateur peut permettre de passer continuellement du domaine spatial au domaine spectral.

En dehors de la mécanique quantique, l'intégralité des travaux réalisés sur ce domaine se sont limités à une transformée d'ordre α réel; dans tout ce qui suit, nous considérerons uniquement $\alpha \in \mathbb{R}$. Dans ce cas, on peut remarquer que l'intégrale de (1) est liée à la transformée de Fourier de: $f(x) \exp(\pm i \cot \alpha x^2 / 2)$.

Etudions à présent comment se comporte la Tff vis à vis des opérateurs de translation et de modulation. On a:

$$F_\alpha[T_{-y}.f](x) = \exp\left[\frac{i}{2} y^2 \sin \alpha \cos \alpha + ixy \sin \alpha\right] F_\alpha[f](x + y \cos \alpha) \quad (3)$$

Le module de Tff n'est donc pas invariant par translation comme l'est celui de la transformée de Fourier. Nous serons amenés à exploiter cette propriété dans la partie consacrée aux applications.

De plus,

$$F_\alpha[D_a.f](x) = \left(\frac{1 - i \cot \alpha}{a^2 - i \cot \alpha}\right)^{1/2} \exp\left[\frac{i}{2} x^2 \cot \alpha \left(1 - \frac{\cos^2 \beta}{\cos^2 \alpha}\right)\right] F_\beta[f]\left(\frac{x \sin \beta}{a \sin \alpha}\right) \quad (4)$$

où β est défini par $\cot \alpha = a^2 \cot \beta$. La dilatation modifie donc profondément le comportement de la Tff, et nous verrons une conséquence de ce phénomène lors de l'étude des algorithmes.

3. ALGORITHMES DE CALCUL DE LA TFF

Il existe un nombre important de travaux proposant une implémentation digitale de la transformée de Fourier fractionnaire. Les plus intéressants sont, à notre sens, ceux basés sur la FFT^[3] (Fast Fourier Transform). Nous débutons cette partie par quelques rappels sur la transformée de Fourier discrète (TFD). Nous détaillerons ensuite les algorithmes que nous avons codés.

3.1. La Transformée de Fourier Discrète.

On considère 2 suites de nombres complexe $x(n)$ et $X(k)$ périodiques de période N . La TFD et la TFD inverse peuvent alors s'exprimer de la façon suivante:

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-i2\pi nk/N} \quad x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{i2\pi nk/N} \quad (5)$$

On réalise ainsi le calcul de la TFD sur un nombre finit d'échantillons.

Le nombre d'opérations requise pour calculer un coefficient de Fourier est de l'ordre de N et l'algorithme de la transformée est alors en $O(N^2)$. Lorsque N est une puissance de 2 ($N=2^m$), il existe des algorithmes rapides de calcul basés sur la périodicité de l'exponentiel complexe. Il existe différentes versions de ces algorithmes; l'une d'entre elle résulte de la factorisation:

$$\sum_{n=0}^{N-1} x(n) e^{-i2\pi nk/N} = \sum_{n=0}^{N/2-1} x(2n) e^{-i2\pi nk/(N/2)} + W^k \sum_{n=0}^{N/2-1} x(2n+1) e^{-i2\pi nk/(N/2)}$$

En effet, cette réécriture de (5) permet d'effectuer le calcul de la TFD de N points en faisant 2 TFD de N/2 points. En itérant ce procédé, on peut réaliser une transformée de Fourier discrète en $O(N \log N)$ opérations.

Ce calcul de la transformée de Fourier discrète s'effectue sur le domaine spatiale $[0, N-1]$ vers le domaine spectral $[0, N-1]$. Le domaine spectral étant périodisé, les coefficients $X(k)$, $k \in [N/2; N-1]$ sont la répétitions des $X(k)$ pour $k \in [-N/2, -1]$. Afin d'observer la TFD centrée en 0, c'est-à-dire avec les fréquences négatives à gauche, il convient de réordonner les coefficients. Nous nomerons ce procédé FFTShift.

Le domaine spatial étant périodisé, on calcule en réalité la Tff de la fonction translatée de $N/2$. Dans le cas de Fourier, seul la phase est affecté mais cela fausse totalement les calculs lorsque l'on compose cet algorithme avec d'autres opérations comme dans le cas du calcul de la Tff.

La démarche suivit pour la TFD sera donc la suivante: une FFTShift, une FFT, et une FFTShift de nouveau. Dans le cas bidimensionnel, La FFTShift correspond à l'inversion des quarts d'images.

3.2. Algorithmes Matriciels.

➤ Description matricielle de la diffraction à distance finie

Dans l'article [4], il est proposé de calculer la Tff digitale en multipliant le signal d'entrée (vecteur colonne) par la matrice:

$$F_{\varepsilon} = \begin{bmatrix} 1 & \dots & W^{(1/2)m^2} & \dots & W^{(1/2)(N-1)^2} \\ \vdots & & \vdots & & \vdots \\ W^{(1/2)n^2(1-\varepsilon^2)} & \dots & W^{(1/2)[(n-m)^2-n^2\varepsilon^2]} & \dots & W^{(1/2)[(n-(N-1))^2-n^2\varepsilon^2]} \\ \vdots & & \vdots & & \vdots \\ W^{(1/2)(N-1)^2(1-\varepsilon^2)} & \dots & W^{(1/2)[(N-1-m)^2-(N-1)^2\varepsilon^2]} & \dots & W^{-(1/2)(N-1)^2\varepsilon^2} \end{bmatrix}$$

Avec $\varepsilon = \sin\phi$. Dans le cas bidimensionnelle, si U est la matrice de l'image, cette opération se réalise de la manière suivante: $U\varepsilon = F_{\varepsilon} U F_{\varepsilon}^t$ où F_{ε}^t est la matrice transposée de F_{ε} .

La pré-multiplication réalise la transformée sur les lignes de l'image, alors que la post-multiplication agit sur les colonnes.

Nous avons codé cet algorithme pour comparer les temps de calcul avec ceux basés sur la FFT. Il apparait que cet algorithme souffre plusieurs défauts:

-Pour $\varepsilon=0$, la matrice en question n'est pas la matrice identité mais la matrice de Fresnel. Cette propriété est une conséquence du changement de variable effectué dans les formules conduisant à F_{ε} . Nous reviendrons sur ce problème pour les algorithmes basés sur la FFT.

-Pour $\varepsilon=1$, on obtient la matrice de Fourier F multipliée par un facteur de phase quadratique:

$$\begin{bmatrix} 1 & \dots & 1 & \dots & 1 \\ \vdots & & \vdots & & \vdots \\ 1 & \dots & W^{-nm} & \dots & W^{-n(N-1)} \\ \vdots & & \vdots & & \vdots \\ 1 & \dots & W^{-(N-1)m} & \dots & W^{-(N-1)^2} \end{bmatrix} \times \begin{bmatrix} 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & W^{(1/2)m^2} & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & W^{(1/2)(N-1)^2} \end{bmatrix}$$

ceci vient du fait qu'il manque une transparence de courbure^[5,6] pour obtenir une phase correcte.

-Les puissances de W sont numérotées de 0 à N-1 ce qui déforme le résultat et déplace l'image (voir les images dans la partie 4). Une amélioration immédiate consiste à recentrer l'algorithme en numérotant les puissances de W de $-N/2$ à $N/2-1$.

La composition n'est bien sur pas vérifiée mais, de toutes façons, aucun algorithme ne respecte parfaitement cette propriété en dehors de l'inversion de la transformée.

➤ **“Puissance” de F.**

Considérons F_α au sens des développements limités des fonctions exponentielles et logarithmes. Celle-ci respecte les conditions suivantes: $F^0=I$, $F^1=F$, $F^\alpha F^\beta = F^{\alpha+\beta}$ et $F^\alpha F^{-\alpha}=I$, de sorte qu'elle apparait naturellement comme un candidat au calcul de la Tff digitale.

En fait, une telle implémentation fournit des résultats très différents de la Tff continue (voir réf [7]). Néanmoins, la recherche d'une "puissance fractionnaire" de la matrice de Fourier est un point de départ pour un bon nombre de travaux proposant une implémentation digitale de la Tff. La difficulté d'une telle recherche résulte du fait qu'il n'y a pas unicité des antécédents matriciels de l'équation $A^p=F^q$; $F^{q/p}$ n'est donc pas unique.

Nous n'entrerons pas plus dans les détails de ces algorithmes dont nous sommes très rapidement désintéressés. Ceux-ci sont trop coûteux en temps de calcul ($O(N^2)$) pour fournir une implémentation efficace.

3.3. Algorithmes de calcul basés sur la FFT.

➤ **Calcul numérique de la Tff à l'aide d'une unique FFT.**

Cet algorithme exposé dans [8] décompose la Tff de façon semblable au point de vu en l'optique métaxiale. La méthode consiste à multiplier le signal d'entrée par un facteur de phase quadratique, effectuer une FFT, et multiplier de nouveau le résultat par un facteur de phase quadratique.

L'intégrale simple qui décrit la Tff d'une fonction peut être considérée comme la transformée de Fourier du produit de cette fonction par un facteur de phase quadratique, comme le montre la définition suivante:

$$F^{(p)}[u_0(x_0)] = \int u_0(x_0) \exp[i\pi(x_p^2 + x_0^2)/(s^2 \tan \phi)] \exp[-i2\pi x_p x_0/(s^2 \sin \phi)] dx_0$$

Où l'on néglige un facteur de phase constant. On a $\phi = p \frac{\pi}{2}$ et $s = \sqrt{\lambda f_1}$, où s est un facteur d'échelle. Après un changement de variable:

$$F^{(p)}[u_0(sx \sin \phi)] = U(\xi) = s \sin \phi \int u_0(sx \sin \phi) \exp[i\pi(\xi^2 + x^2 \sin^2 \phi) / \tan \phi] \exp(-i2\pi \xi x) dx$$

$$\text{soit } U(\xi) = s \sin \phi \exp[i\pi \xi^2 / \tan \phi] \int u(x) \exp[i\pi x^2 \sin \phi \cos \phi] \exp(-i2\pi \xi x) dx$$

$$\text{avec } \xi = \frac{x_p}{s}, \quad x = \frac{x_0}{s \sin \phi}, \text{ et } u(x) = u_0(sx \sin \phi)$$

La forme discrète de $U(\xi)$ est alors

$$U(k \delta \xi) = s \delta x \sin \phi \exp\left[\frac{i\pi k^2 \delta \xi^2}{\tan \phi}\right] \sum_{n=-N/2}^{(N/2)-1} u(n \delta x) \exp(-i2\pi n k \delta \xi \delta x + i\pi n^2 \delta x^2 \sin \phi \cos \phi) \quad (6)$$

où $x = n \delta x$ et $\xi = k \delta \xi$. δx et $\delta \xi$ sont les périodes d'échantillonnage dans les domaines spatiales et fréquentielles, et n et k sont des entiers. En introduisant le nombre d'échantillon N, la longueur du domaine d'échantillonnage Δx , avec $\delta x = \Delta x/N$, et utilisant le fait que $\delta \xi = 1/\Delta x$ (théorie de la TFD), On peut réécrire (6) comme suit:

$$U_k = \frac{s \Delta x \sin \phi}{N} \exp[i\pi k^2 / (\Delta x^2 \tan \phi)] \sum_{n=-N/2}^{(N/2)-1} \tilde{u}_n \exp(-i2\pi n k / N)$$

$$U_k = \frac{s \Delta x \sin \phi}{N} \exp[i\pi k^2 / (\Delta x^2 \tan \phi)] TFD\{\tilde{u}_n\}$$

$$\text{avec } \tilde{u}_n = u(n \Delta x / N) \exp[i\pi n^2 \Delta x^2 \sin 2\phi / (2N^2)]$$

Finalement,

$$U_k = \frac{\Delta x_0}{N} \exp \left[\frac{i \pi k^2 s^2 \sin 2\phi}{2 \Delta x_0^2} \right] \sum_{n=-N/2}^{(N/2)-1} \tilde{u}_n \exp \left[\frac{i \pi n^2 \Delta x_0^2}{N^2 s^2 \tan \phi} \right] \exp(-i 2 \pi n k / N) \quad (7)$$

La période d'échantillonnage pour que le facteur de phase quadratique satisfasse le théorème de Nyquist pour x_0 est la suivante:

$$\delta x_0 \leq \frac{s^2 \tan \phi}{\Delta x_0}. \text{ Pour } x_p, \text{ on a } \delta x_p \leq \frac{s^2 \tan \phi}{\Delta x_p}$$

Par ailleurs, on a $\Delta x_p = \frac{N s^2 \sin \phi}{\Delta x_0}$, $\delta x_p = \frac{s^2 \sin \phi}{\Delta x_0}$ et $\Delta x_0 = N \delta x_0$.

Le nombre d'échantillon doit donc respecter la condition suivante:

$$\frac{\Delta x_0^2}{s^2 \tan \phi} \leq N \leq \frac{2 \Delta x_0^2}{s^2 \sin 2\phi} \quad (8)$$

Si l'on choisit de fixer le paramètre N afin de permettre l'utilisation des algorithmes de FFT ($N=2n$), les conditions sur la longueur du domaine d'échantillonnage Δx_0 s'écrivent:

$$\left(\frac{N s^2 \sin 2\phi}{2} \right)^{1/2} \leq \Delta x_0 \leq (N s^2 \tan \phi)^{1/2} \quad (9)$$

Il faut noter que le domaine d'échantillonnage est normalement plus grand que le domaine de la fonction. Si c'est le cas, on considère simplement la fonction nulle en dehors de son domaine de définition. Pour un ordre suffisamment proche de 1, le facteur de phase quadratique oscille lentement et les critères établissant un domaine d'échantillonnage correcte doivent être basés sur la fonction elle-même. Pour un ordre proche de 0, Δx_0 est trop petit (éq (9)) et N trop grand (éq (8)). Néanmoins, on peut s'affranchir de cette difficulté en calculant, dans un premier temps, une transformée fractionnaire d'ordre $p'=1+p$, suivit ensuite d'une TFD inverse.

➤ Calcul numérique de la FRT via 2 FFT.

Un autre algorithme utilisant 2 FFT (réf [9]) possède des caractéristiques intéressantes. Cette méthode est basée sur l'implémentation optique de type II du système de Lohmann^[10]. L'échantillonnage est identique dans chaque domaine fractionnaire:

$$\delta x_0 = \delta x_p = \tilde{\delta x} = \sqrt{\frac{\lambda f_1}{N}} \quad \text{et} \quad \Delta x_0 \Delta x_p = \lambda f_1 N$$

L'algorithme se réalise en trois temps:

1) L'action de la première lentille produit la multiplication du signal d'entrée par le facteur de phase quadratique:

$$\exp \left[-i \frac{\pi}{N} \tan(p \frac{\pi}{4}) m^2 \right]$$

2) L'action de la propagation de l'onde dans l'espace libre produit la multiplication par un facteur de phase dans le domaine de Fourier, qui peut s'effectuer de la façon suivante:

Réaliser la TFD et multiplier le signal obtenu par le facteur de phase quadratique:

$$\exp \left[-i \frac{\pi}{N} \sin(p \frac{\pi}{2}) \tilde{m}^2 \right].$$

réaliser la TFD inverse.

3) L'action de la seconde lentille produit à nouveau une multiplication par le facteur de phase quadratique:

$$\exp \left[-i \frac{\pi}{N} \tan(p \frac{\pi}{4}) m'^2 \right]$$

Les conditions d'échantillonnage s'écrivent: $\left| \sin\left(p \frac{\pi}{2}\right) \right| \leq 1$, qui est toujours vérifiée, et

$\left| \tan\left(p \frac{\pi}{4}\right) \right| \leq 1$, qui est vérifiée lorsque $|p| \leq 1$.

Cet Algorithme permet donc l'implémentation de la Tff digitale pour un ordre $p \in [-1;1]$, ce qui est suffisant pour n'importe quel ordre réel étant données les propriétés de symétrie de la Tff.

Notons qu'un facteur de phase constant a été négligé et que l'on doit multiplier le résultat par la constante $M_p = \exp\left\{-i\pi \operatorname{sgn}\left[\sin\left(p \frac{\pi}{2}\right)\right] / 4 + ip \frac{\pi}{4} + i \frac{\pi}{4}\right\}$ si l'on souhaite une identification totale.

➤ **Comparaison des algorithmes.**

Notons d'abord que, en terme de temps de calcul, le premier algorithme est sensiblement plus avantageux puisque l'on effectue 1 seule FFT. Ceci n'est plus valable lorsque l'on choisit un ordre proche de 0 et que l'on effectue une Tff d'ordre $1+p$ suivit d'une FFT inverse.

Il est nécessaire de faire une adéquation entre la solution analytique et le résultat digital pour chacun des algorithmes. Lorsque $\Delta x_0 = (Ns^2 \tan \phi)^{1/2}$ (cas limite de l'équation (9)), il apparaît que le premier algorithme produit un résultat très proche de la solution analytique, et en tout cas nettement meilleur que celui du second algorithme. Ceci provient du fait que la valeur de Δx_0 est un multiple entier de la plus petite période du facteur de phase quadratique.

Lorsque la taille du domaine d'échantillonnage n'est pas égal à sa valeur optimale, les 2 algorithmes produisent des résultats semblables. Dans ce cas, les résultats sont d'autant plus mauvais que l'ordre diminue puisque les oscillations du facteur de phase sont de plus en plus rapides. Dans tout les cas, les écarts à la solution analytique sont réellement significatifs au niveau de la phase alors que le module est toujours satisfaisant. Par ailleurs, la précision de la méthode utilisant une Tff augmente de nouveau en s'approchant de 0 si l'on utilise une FFT inverse consécutivement à une Tff d'ordre $1+p$.

Les deux méthodes permettent d'inverser la transformée, mais alors que la seconde ne pose aucun problème, la première ne s'inverse pas, dans le cas général, en réalisant une Tff d'ordre $-p$. En revanche, pour cet algorithme, lorsque l'on choisit $\Delta x_0 = (Ns^2 \tan \phi)^{1/2}$ et p_1+p_2 entier (1,2,3...), la composition d'une Tff d'ordre p_1 et d'une Tff d'ordre p_2 conduit à un résultat exact. La distribution de la phase et du module est alors parfaitement identique à la solution analytique.

La méthode utilisant une FFT apparaît jusqu'ici nettement meilleur que la seconde. En fait celle-ci est très problématique en traitement du signal parce que l'on est en présence de paramètres supplémentaires que l'on ne peut pas choisir indépendamment du signal. En outre, et c'est là son principal défaut, le changement de variable effectué dans les formules produit une dilatation de la fonction dans le domaine fractionnaire. Nous avons vu (formule (4)) que la dilatation modifie fortement le résultat, et notamment l'ordre de la transformée. Ainsi, à échelle constante, on ne peut pas comparer deux Tff d'ordre différent parce que la fonction a été dilatée différemment.

Pour la seconde méthode, l'échantillonnage est identique dans chaque domaine fractionnaire et tous les résultats sont cohérents entre eux quelque que soit l'ordre. Cette propriété en fait un outil nettement plus appréciable dans un contexte purement digital. Dans la quatrième partie, nous illustrerons par des images cet écart entre les deux algorithmes.

3.4. Utilisation des algorithmes.

Nous proposons dans ce qui suit quelques algorithmes inédits découlant directement des deux précédents.

➤ **Cas particulier de l'algorithme à une FFT.**

Pour notre programme, nous désirons implémenter la méthode à une FFT sans que l'utilisateur n'ait à se préoccuper d'autre paramètre que l'ordre de la transformée. De plus, s et Δx_0 ont

tout deux une signification physique qui disparaît dans le contexte digital (par exemple une image quelconque numérisée).

Il serait avantageux de palier aussi au problème de changement d'échelle, par exemple en interpolant le signal par les fonctions sinus cardinale. Il existe des algorithmes rapides et relativement souple (réf [11]) pour ce type d'interpolations mais plusieurs difficultés apparaissent, notamment le fait que nos algorithmes ne fonctionnent que pour des puissances de 2 échantillons. Nous n'avons pas été plus loin dans cette direction mais il s'agit là d'une ouverture intéressante.

Dans l'équation (7), on peut remarquer que s et Δx_0 sont toujours en rapport et l'on peut fixer arbitrairement $s=1$ sans perte de généralité. Reste alors à déterminer Δx_0 . En imposant que la Tff d'ordre $-p$ inverse parfaitement une Tff d'ordre p , la formule (7) nous conduit à choisir Δx_0 tel que:

$$\frac{\sin 2\phi}{2\Delta x_0^2} = \frac{\Delta x_0^2}{N^2 \tan \phi} \quad \text{On a alors: } \Delta x_0^2 = |N \sin \phi|$$

Cette valeur de Δx_0 vérifie toujours la relation (7). L'algorithme ainsi réalisé ne dépend plus que d'un unique paramètre qui est l'ordre de la transformée. Un autre intérêt de cette méthode est que, en plus d'inverser la transformation, la composition de deux Tff d'ordre p_1 et p_2 conduit toujours à un résultat exact lorsque p_1+p_2 est un entier.

➤ **Le cas bidimensionnel: la Tff locale.**

On définit généralement la transformée de Fourier fractionnaire en utilisant le même ordre dans les deux directions. Cependant, il est possible (réf [12]) de définir cette transformée avec un ordre différent dans la direction horizontale et verticale

Le kernel de cette transformée est alors obtenue par le produit tensoriel de deux kernel unidimensionnels:

$$F^{a_x, a_y} [\{f(u, v)\}](u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} B_{a_x}(u, u') B_{a_y}(v, v') f(u', v') du' dv'$$

$$\text{Avec } B_a(u, u') = \frac{\exp[-i(\pi\tilde{\phi}/4 - \phi/2)]}{\sqrt{\sin \phi}} \exp[i\pi(u^2 \cot \phi - 2uu' / \sin \phi + u'^2 \cot \phi)] \text{ pour}$$

$0 < |a| < 2$, où $\phi = a\pi/2$ et $\tilde{\phi} = \text{sgn}(\sin \phi)$. On a aussi $B_0(u, u') = \delta(u - u')$ et

$B_2(u, u') = \delta(u + u')$. Il suffit donc d'effectuer l'algorithme en ligne avec un ordre a_x , puis en colonne avec un ordre a_y .

En fait, il est possible d'implémenter un cas plus général que le précédent: La Tff 2D se réalise successivement sur chaque ligne et chaque colonne de l'image. Il est donc possible de calculer une Tff dont l'ordre varie spatialement, mais dont la composante horizontale est identique sur une même ligne et dont la composante verticale est identique sur une même colonne, gardant ainsi le bénéfice des algorithmes rapides de calcul.

Cela nous conduit à une transformée de Fourier fractionnaire locale, c'est-à-dire une transformée dont l'ordre dépend de la position. Cet ordre peut varier arbitrairement d'un échantillon à l'autre pourvu que cette variation s'effectue uniquement dans les directions horizontale ou verticale.

➤ **Le cas monodimensionnel: transformée de Radon de la distribution de Wigner.**

La remarque précédente d'un ordre évoluant spatialement nous a conduit à une autre application de ces algorithmes ne concernant que le domaine digital.

On considère un signal monodimensionnel f que l'on recopie identiquement sur chaque ligne d'une l'image. On réalise alors l'algorithme précédent en incrémentant linéairement l'ordre α dans la direction verticale de α_1 (première ligne) à α_2 (dernière ligne).

Ainsi, on peut suivre l'évolution d'un signal dans chaque domaine fractionnaire en un coup d'oeil, ce qui, en soit, présente un intérêt certain. En outre, on peut déduire que cette représentation n'est autre que celle de la transformée de Radon de la distribution de Wigner de f de l'angle α_1 à α_2 .

L'ordre étant variable, il est difficile d'utiliser l'algorithme à une Tff car l'échantillonnage varie d'un ordre à l'autre et la comparaison est faussée. Nous verrons par la suite quel est l'intérêt de ces algorithmes.

4. IMPLEMENTATION

L'objectif principal de ce projet était la réalisation d'un logiciel de travail sur la transformée de Fourier fractionnaire digitale. Nous avons codés un certain nombre d'outils sous environnemnt visual C⁺⁺. Ce programme réalise les opérations basiques de tout logiciel d'images (ouvrir, sauver, classer les images...). Nous avons utilisé deux librairies: Matrox pour l'aspect gestion graphique et m⁺⁺ comme librairie mathématique.

Dans ce qui suit, nous présentons uniquement les fonctions du menu traitements. Les images sont celles des boites de dialogue associées. Nous montrerons ensuite quelques exemples d'images réalisées avec ce programme.

4.1. Présentation.

➤ Negative Image.

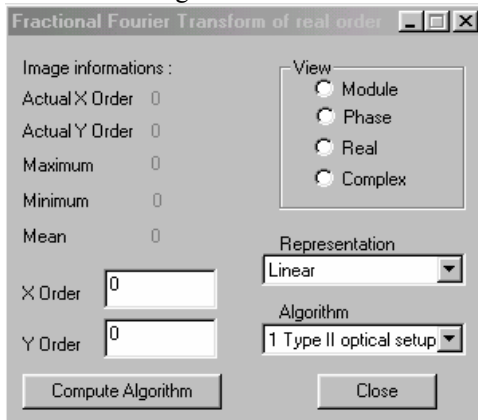
On sait que l'oeil repond logarithmiquement à l'intensité de la lumière, et est donc plus sensible dans les teintes claires. L'inversion de l'image permet donc, dans certains cas, d'améliorer la visualisation. Sachant que l'on a 256 niveaux de gris, on réalise cette opération en associant au niveau de gris z le niveau de gris $255-z$.

➤ Fourier transform / inverse Fourier transform.

Nous avons utilisé la FFT de la librairie m⁺⁺, qui a l'avantage d'être particulièrement rapide. Le problème de cette méthode est que l'on utilise la FFT comme une boîte noire et l'on est ainsi tributaire de sa qualité. Il arrive notamment que la phase de certaines implémentations ne soit pas fiable (réf [13]), mais un certain nombre de test nous a révélé que celle-ci est de bonne qualité. La normalisation est celle des formules (5) afin d'être identique à celle que produisent les algorithmes de calcul de la FRT lorsque l'on choisit un ordre égal à 1. On représente la TFD avec la visualisation logarithmique et la TFD inverse avec la visualisation linear (voir paragraphe suivant).

➤ Fractional Fourier Transform.

Il s'agit d'une boîte de dialogue permanente qui est associée à un document. Elle permet d'effectuer la Tff avec un ordre horizontal et un ordre verticale que l'on rentre dans les "edit box" en bas à gauche.



La première étape est le choix de l'algorithme dans la "combo box" (en bas à droite) parmi les suivants: Algorithme à 1FFT, Algorithme à 2 FFT et l'implémentation matricielle centrée ou non.

La largeur et la hauteur de l'image doivent être des puissances de 2 pour permettre l'utilisation de la FFT. L'ordre doit être compris entre -1 et 1 (un ordre 2 fonctionne également) pour l'algorithme à 2 FFT et entre -2 et 2 pour celui à 1 FFT.

Le signal complexe obtenu $|A(k,l)|\exp(i\phi(k,l))$ est stockée dans un tableau de données où les partie réelle et imaginaire sont des doubles. On note max la valeur maximale du module des valeurs de ce buffer.

Dans le coin supérieur droit, on choisit de visualiser le module, la phase, la partie réelle ou la partie imaginaire du signal complexe associé à l'image, en cliquant sur le "radio button" associé. Les valeurs sont représentées sur 256 niveaux de gris selon la représentation choisit dans la "combo box" en bas à droite.

Pour la phase, on dispose de deux représentations:

- Phase absolue: $\phi \in]-\pi; \pi] \rightarrow \phi' = \frac{|\phi|}{\pi} \times 255$

- Phase Totale: $\phi \in]-\pi; \pi] \rightarrow \phi' = \frac{\phi + \pi}{2\pi} \times 255$

La visualisation du module, de la partie réelle et de la partie imaginaire fonctionne avec les quatre représentations suivantes:

- Linear: $A'(i, j) = 255 \times \frac{|A(i, j)|}{\max} \quad \max \neq 0$

- Logarithmic: $A'(i, j) = 255 \times \frac{\ln(|A(i, j)| + 1)}{\ln(\max + 1)}$

- Truncate: $A'(i, j) = \begin{cases} 0 & \text{si } A(i, j) < 0 \\ 255 & \text{si } A(i, j) > 255 \\ A(i, j) & \text{si non} \end{cases}$

- Truncate Abs: $A'(i, j) = \begin{cases} 255 & \text{si } |A(i, j)| > 255 \\ |A(i, j)| & \text{si non} \end{cases}$

Notons toutefois que Truncate et Truncate Abs sont identiques pour le module puisque celui est forcément positif.

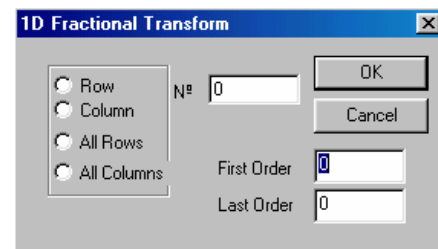
En haut à gauche, à chaque changement de visualisation, la boîte de dialogue fournit les renseignements suivants:

- L'ordre actuel en X (horizontal) et en Y (vertical) du signal, permettant de savoir dans quel domaine fractionnaire on se trouve.
- Les valeurs maximale et minimale de l'image visualisée (entre $-\pi$ et π pour la phase et entre $-\infty$ et $+\infty$ pour les autres)
- La moyenne des valeurs de l'image.

➤ 1D Fractional Fourier Transform.

Les deux premiers choix "row" et "column" produisent l'algorithme décrit au paragraphe « Le cas monodimensionnel: transformée ... », respectivement sur une ligne ou une colonne. La ligne ou la colonne se choisie en entrant sa position dans l'edit Box du haut (0 correspond à la ligne du haut ou à la colonne la plus à gauche de l'image).

Le choix "all rows" (respectivement "All columns") réalise une TFF horizontale (respectivement verticale) locale en incrémentant l'ordre de la transformée d'une ligne (respectivement colonne) à l'autre.



Dans les deux cas, le pas δ_p est incrémenté linéairement de First Order à Last Order. Si ces deux ordres sont identiques, l'ordre est constant sur chaque ligne ou chaque colonne.

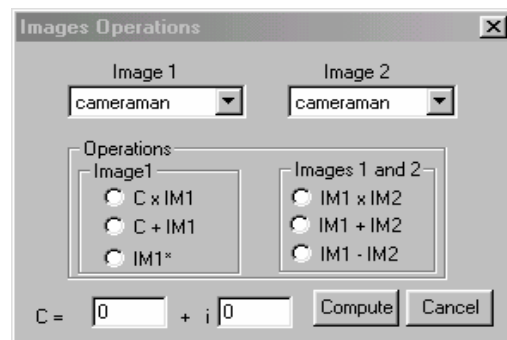
➤ Images operations

Les deux "combo box" contiennent chacune la liste de tout les documents ouverts. On élit ainsi deux images: l'image 1 à gauche et l'image 2 à droite.

Le choix de l'opération à réaliser s'effectue en cliquant dans les "radio button". Ceux de la colonne de gauche agissent uniquement sur l'image 1.

On considère un scalaire complexe sous la forme $C=a+ib$. On a dans l'ordre:

- $C \times IM1$ qui produit la multiplication du buffer complexe associé à l'image 1 par la constante C
- $C + IM1$ qui produit l'addition de la constante C au buffer complexe associé à l'image 1.
- $IM1^*$ qui transforme les nombres complexes du même buffer en leur conjugué.



Ceux de la colonne de droite dépendent des 2 images:

- $IM1 \times IM2$ réalise le produit des 2 buffers complexes associés aux images 1 et 2.

- $IM1 + IM2$ réalise l'addition des 2 buffers complexes associés aux images 1 et 2.

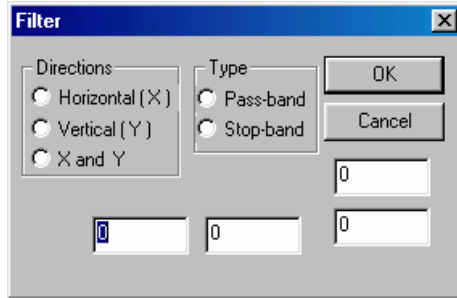
- $IM1 - IM2$ réalise la soustraction des 2 buffers complexes.

Dans tous les cas, le résultat est stocké dans le buffer complexe associé à l'image 1 et écrase les données précédentes.

➤ Phase Filter

Ce traitement réalise la division du buffer complexe d'un signal par son module, transformant ainsi le signal en sa phase: $|A(x)|e^{i\phi(x)} \rightarrow e^{i\phi(x)}$ L'intérêt de cet outil est pour la corrélation.

➤ Filters.



Il s'agit là simplement de filtre coupe-bande ou passe bande (on choisit le type avec l'un des deux "radio button" de droite) qui forcent les valeurs du buffer complexe à 0. Les trois "radio button" à gauche permettent de choisir la direction du filtre: "horizontal" correspond au deux edits de gauche (coordonnées en X) et "vertical" au deux edits de droite (coordonnées en Y). "X and Y" réalise en une fois les deux filtrages précédents.

Par exemple, un filtre coupe-bande horizontal supprime tous les coefficients de l'image situés entre les deux valeurs de X entrées dans les deux edits de gauche.

4.2. Exemples d'images.

Considérons les deux images d'un E (fig. 4); l'un est centré et l'autre est translaté vers le bord de l'écran. Le module de leur TFD est identique (fig. 4c) alors que la phase est différente.

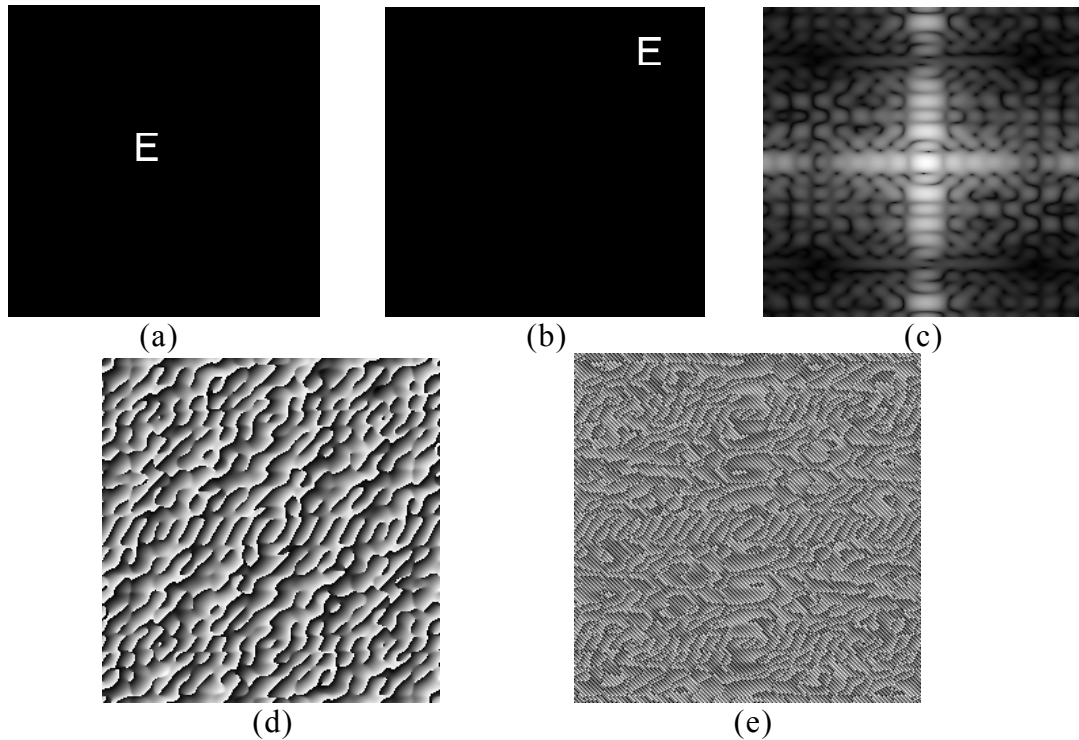


Fig. 4: (a) Image d'un E centré. (b) Image d'un E décentré. (c) Module de la TFD des image (a) et (b) en visualisation logarithmic. (d) Phase de la TFD de (a) en visualisation Phase totale. (e) Phase de la TFD de (b).

La figure 5 représente les transformées de Fourier fractionnaires d'une fonction "porte" pour différents ordres, effectuées avec l'algorithme utilisant deux FFT. On voit apparaître plusieurs motifs de diffraction; l'énergie se délocalise à mesure que l'ordre augmente. Le temps de calcul est d'entre 3 à 4[s] sur un Pentium II (300 MHz et 64 Mega de RAM) avec 256 échantillons.

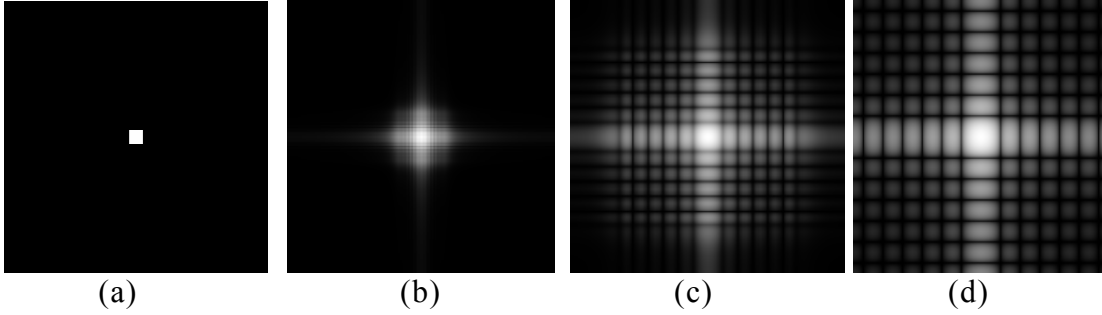


Fig. 5: (a) Image d'une fonction porte. (b) Tff d'ordre 0.1 de (a) en visualisation logarithmique (pour l'algorithme à deux FFT). (c) Tff d'ordre 0.5 de (a). (d) Tff d'ordre 1 de (a).

Sur les figures suivantes, on observe le module de la Tff d'une image test pour un ordre 0.01. L'algorithme à deux FFT respecte la continuité avec l'opérateur identité alors que celui à une FFT produit une transformée dont l'ordre semble beaucoup plus élevé. Néanmoins, on retrouve un résultat comparable en effectuant une Tff d'ordre 1.01 suivie d'une TFD inverse (Fig. 6d). Le temps de calcul est environ de 2[s] pour l'algorithme à une FFT.

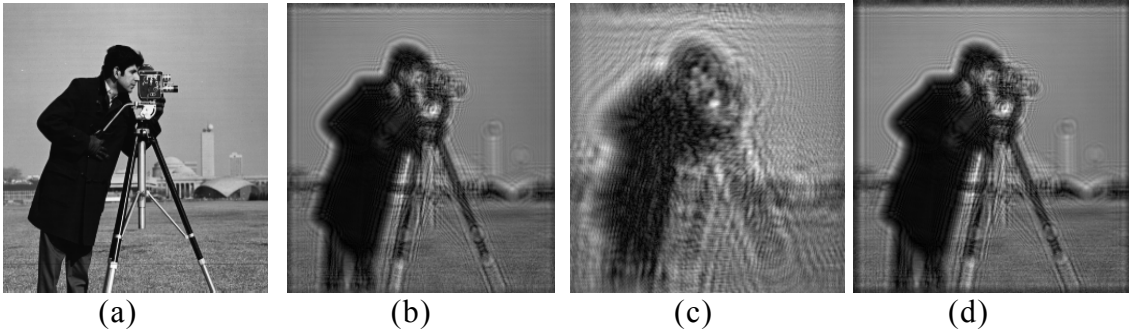


Fig. 6:(a) Image cameraman. (b) Module de la Tff d'ordre 0.01 en visualisation linéaire pour l'algorithme à deux FFT. (c) Module de la Tff d'ordre 0.01 pour l'algorithme à une FFT. (d) Module de la Tff d'ordre 1.01 suivit d'une TFD inverse pour l'algorithme à une FFT.

Nous avons parlé auparavant de certains cas de composition parfaite pour la méthode à une FFT. Ceci est illustré sur la figure 7 où les ordres ont été choisis tel que $p_1 + p_2 = 2$. La Tff d'un ordre 2 produit bien l'inversion de l'image.

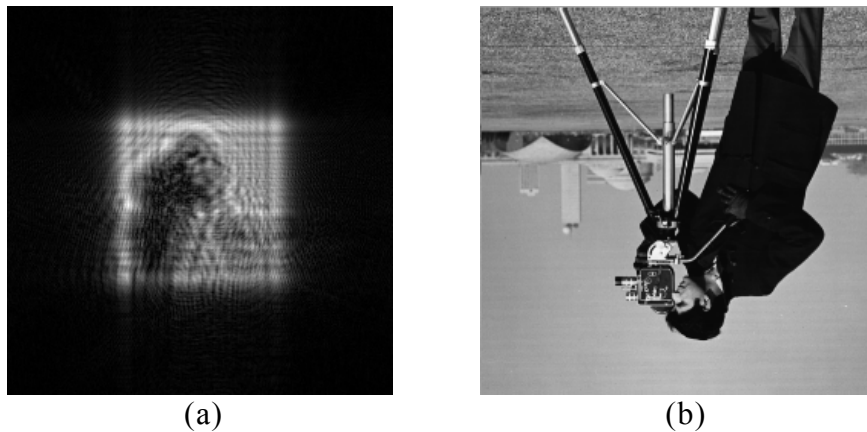


Fig. 7: (a) Module de la Tff d'ordre 0.6 de cameraman pour l'algorithme utilisant une FFT en visualisation linear. (b) Partie réelle de la Tff d'ordre 2 de cameraman en ayant composé une Tff d'ordre 0.6 et une Tff d'ordre 1.4 avec la méthode utilisant une FFT.

Voici la représentation de la transformée de Radon de la distribution de Wigner d'un angle 0 à $\pi/2$ pour quelques fonctions (Fig. 8); L'algorithme à deux FFT a été utilisé. On peut observer

comment évolue un Dirac dans le domaine spatial vers une constante dans le domaine spectral (inversement, une constante deviendrait un Dirac), et comment une fonction “porte” se change en sinus cardinal. D’après les propriétés de la Tff, cette transition s’effectue continuellement d’un domaine à l’autre.

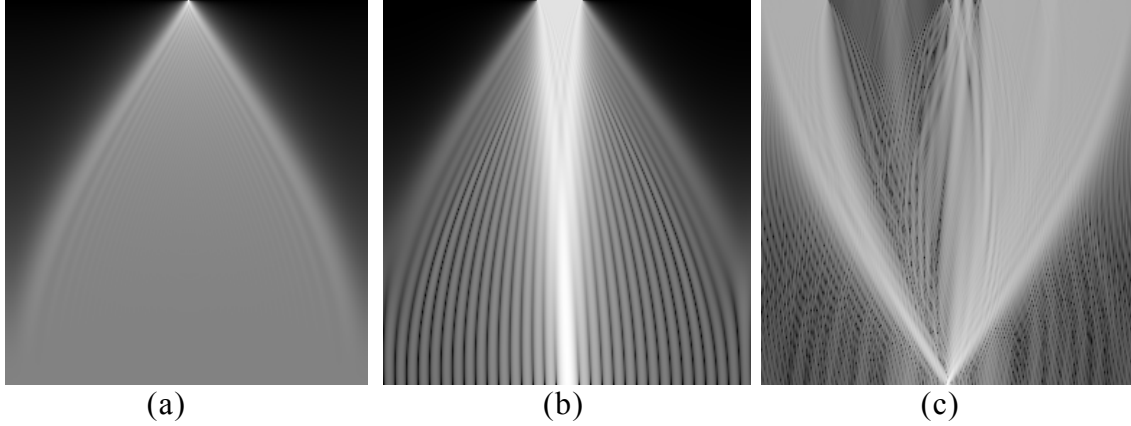


Fig. 8: (a) Transformée de Radon de la WDF d’une distribution de Dirac d’un angle 0 à $\pi/2$ en visualisation logarithmique. (b) Même chose que en (a) pour une fonction porte. (c) Même chose que en (a) pour la 128ème ligne de l’image cameraman.

Enfin, voici le module de la Tff calculé avec l’algorithme matriciel (Fig. 9). Dans les mêmes conditions que précédemment, le temps de calcul est de 45 [s] mais il faut considérer que nos multiplications matricielles ne sont pas optimisées contrairement aux algorithmes de FFT. Le rapport de cette durée avec celle des algorithmes précédent augmente considérablement lorsque la taille de l’image augmente. On observe le recentrage du module en modifiant l’algorithme^[14].

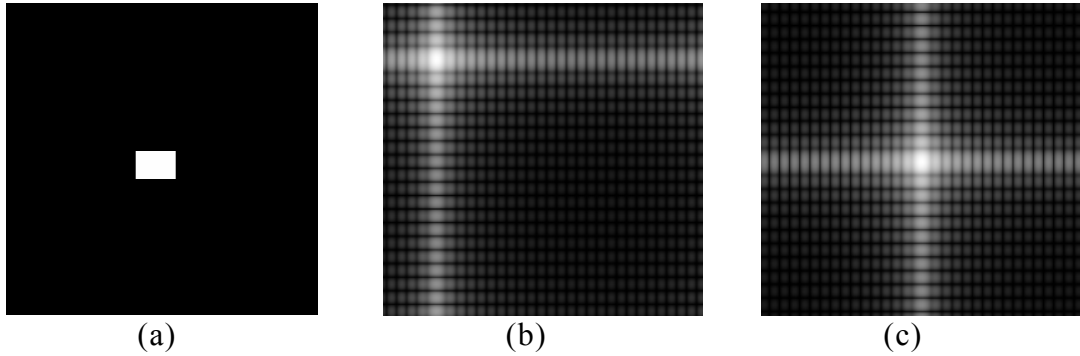


Fig. 9: (a) Image d’une fonction porte. (b) Tff d’ordre 0.6 de (a) en visualisation logarithmique avec l’algorithme matriciel. (c) Tff d’ordre 0.6 de (a) pour l’algorithme matriciel centré.

5. CONCLUSION

Le domaine des mathématiques fractionnaires connaît actuellement un essort important, et la transformée de Fourier fractionnaire en est l’un des thèmes les plus avancés. Elle fut cependant d’avantage exploitée dans le domaine de l’optique car elle permet de modéliser un cas relativement général de la propagation d’une onde électromagnétique. Il existe d’ailleurs différentes implémentations optique de cette transformée.

Les deux algorithmes basés sur la FFT présente chacun des qualités et des défauts différents, et le bénéfice de leur utilisation dépend essentiellement du contexte de l’application. Il n’existe pas, à l’heure actuelle, un algorithme aussi satisfaisant que dans le cas de la transformée de Fourier. Le fait que les transformées de Fourier fractionnaires ne se composent pas correctement est un handicap certain, en particulier pour des applications tel que la corrélation fractionnaire qui nécessitent l’enchaînement de deux Tff. Les erreurs sur la phase

sont également problématiques car elles limitent l'analyse de signaux modulés en amplitude et en fréquence telle que celles réalisées avec la transformée en ondelettes ou avec des distributions du type Wigner-Ville.

Nous avons introduit une nouvelle représentation temps-fréquence, la transformée de Radon de la distribution de Wigner, que l'on peut implémenter très simplement à partir des algorithmes précédents. Elle permet notamment de déterminer le domaine fractionnaire optimale pour différent type d'application.

Le domaine de la transformée de Fourier fractionnaire est encore sujet à de nombreuses recherches et son aspect digital est loin d'être entièrement exploré. Le logiciel réalisé peut lui-même être complété par de nouveaux outils de travail facilitant les traitements. D'autres recherches peuvent être également effectuées pour découvrir de nouvelles applications de cette transformée.

6. BIBLIOGRAPHIE

1. V. Namias, "The fractional Fourier transform and its application in quantum mechanics", *J. Inst. Math. Appl.* 25, 241-265 (1980).
2. A.C. McBride and F.H.Kerr. "On Namias's fractional Fourier Transform", *IMA J. Appl. Math* 39, 159-175 (1987).
3. J. W. Cooley, and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Math. Computat.* 19, 297-301 (1965).
4. S. B. Tucker, J. Ojeda-Castañeda and W. T. Cathey, "Matrix description of near-field diffraction and the fractional Fourier transform", *J. Opt. Soc. Am. A/Vol. 16, No 2*, 316-322 (1998).
5. P. Pellat-Finet, "Transformación de Fourier fraccional y propagación del campo electromagnético", *memos de investigación No 121*, Universidad de los Andes, Santafé de Bogotá, (1994).
6. P. Pellat-Finet, "Transfert du champ électromagnétique par diffraction et transformation de Fourier fractionnaire", *C. R. Acad. Sci. Paris, t. 320, Serie IIb*, p. 91-97 (1995).
7. Soo-Chang Pei, Min-Hung Yeh, and Chien-Cheng Tseng, "Discrete Fractional Fourier Transform Based on Orthogonal Projections", *IEEE Transactions on Signal Processing*, Vol. 47, No 5, 1335-1348 (1999).
8. F. J. Marinho, and L. M. Bernardo, "Numerical calculation of fractional Fourier transforms with a single fast-Fourier-transform algorithm", *J. Opt. Soc. Am A* 15, 2111-2116 (1998).
9. J. García, D. Mas, and R. G. Dorsch, "Fractional-Fourier-transform calculation through the fast-Fourier-transform algorithm", *Applied Optics*, vol. 35, 7013-7018 (1996).
10. Adolf Lohmann, "Image rotation, Wigner rotation, and the fractional Fourier transform", *J. Opt. Soc. Am A* 10, 2181-2186 (1993).
11. L.P. Yaroslavsky, "Efficient algorithm for discrete sinc interpolation", *Appl. Opt.* Vol. 36, No 2, 460-463 (1997).
12. A. Sahin, H. M. Ozaktas and D. Mendlovic, "Optical implementation of the two-dimensional fractional Fourier transform with different orders in the two dimensions", *Optics Communications* 120, 134-138 (1995).
13. B. Torrèsani, "Analyse continue par ondelettes", *Savoirs Actuels (interEditions / CNRS editions)*, 1995.
14. E. Cousin, "Transformée de Fourier fractionnaire digitale. Algorithmes et applications ", *Project de fin d'études*, CPE-Lyon, fait à l'Universidad Industrial de Santander, Bucaramanga, 41p. (2000).
15. D. Mendlovic, and H.M Ozaktas, "Fractional Fourier transforms and their optical implementation: I " *J. Opt. Soc. Am. A* 10, 1875-1881 (1993).
16. D. Mendlovic, and H.M Ozaktas, "Fractional Fourier transforms and their optical implementation: II " *J. Opt. Soc. Am. A* 10, 2522-2531 (1993).
17. H.M Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms", *J. Opt. Soc. Am. A* 11, 547-559 (1994).
18. R. G. Dorsh, A. W. Lohmann, Y. Bitran, D. Mendlovic and H. Ozaktas, "Chirp filtering in the fractional Fourier domain", *Applied Optics*, Vol. 33, 7599-7602 (1994).
19. T. Alieva, V. Lopez, F. Agullo-Lopez, L. B. Almeida, "The fractional Fourier transform in optical propagation problems", *Journal of Modern Optics*, Vol. 41, No 5, 1037-1044 (1994).
20. D. Mendlovic, H. M. Ozaktas, B. Barshan and H. Urey, "Space-variant filtering in fractional Fourier domains", *Inst. Phys. Conf. Ser. No. 139: part III*, 285-288 (1995)

21. D. Mendlovic, H. M. Ozaktas and A. W. Lohmann, "Fractional correlation", *Applied Optics*, Vol. 34, 303-309 (1995)
22. D. Mendlovic, Z. Zalevsky, A. W. Lohmann, R. G. Dorsch, "Signal spatial-filtering using the localized fractional Fourier transform", *Optics Communications* 126, 14-18 (1996).
23. D. Mendlovic, Z. Zalevsky, D. Mas, J. García, C. Ferreira, "Fractional wavelet transform", *Appl. Opt.*, Vol. 36, 4801-4806 (1997).
24. Z. Zalevsky, D. Mendlovic, and J. H. Caulfield, "Fractional correlator with real-time control of the space-invariance property", *Appl. Opt.*, Vol 36, No 11, 2370-2375 (1997).
25. Wen-Xiang Cong, Nan-Xian Chen and Ben-Yuan Gu, "Recursive algorithm for phase retrieval in the fractional Fourier transform domain", *Appl. Opt.*, Vol. 37, No 29, 6906-6910 (1998).
26. David Mas Candela, "La transformada de Fourier fraccionària: algorismes i aplicacions en Òptica.", *Tesi Doctoral*, Universitat de València, Departament d'òptica, Juliol 1999.
27. P. Pellat-Finet, "Diffraction entre un émetteur et un récepteur localement toriques. Application à l'étude des systèmes astigmatiques", *C. R. Acad. Sci. Paris*, t.327, Serie II b, 1269-1274 (1999).