An Accurate Discrete Fourier Transform for Image Processing

Normand Beaudoin* and Steven S. Beauchemin†

Abstract

The classical method of numerically computing the Fourier transform of digitized functions in one or in d-dimensions is the so-called discrete Fourier transform *(DFT), efficiently implemented as* Fast Fourier Transform *(FFT) algorithms. In many cases the DFT is not an adequate approximation of the continuous Fourier transform. The method presented in this contribution provides accurate approximations of the continuous Fourier transform with similar time complexity. The assumption of signal periodicity is no longer posed and allows to compute numerical Fourier transforms in a broader domain of frequency than the usual half-period of the DFT. In image processing this behavior is highly welcomed since it allows to obtain the Fourier transform of an image without the usual interferences of the periodicity of the classical DFT. The mathematical method is developed and numerical examples are presented.*

1 Introduction

The Fourier transform and its numerical counterpart, the discrete Fourier transform (DFT), in one or in many dimensions, are used in many fields such as mathematics, physics, chemistry, engineering, image processing, and computer vision [4, 12, 11, 8, 14, 1].

"The DFT is of interest primarily because it approximates the continuous Fourier transform" [4]. In this regard, the DFT, usually computed via a fast Fourier transform (FFT) algorithm, must be used with caution since it is not a correct approximation in all cases [13, 10, 15, 6]. As an example, for a function such as $h(t) = e^{-50t}$, $t \in [0, 1]$, the error on $DFT\{h\}$ around $f = 64$ decreases roughly as $N^{-1/3}$. Hence, one must increase N by a factor of 1000 to decrease the error by a factor of 10.

One may increase the accuracy of the numerical Fourier transform when the number of sampled data points is limited. This can be implemented through the assumption that the function from which the sampled data points are extracted and its derivatives are continuous. In a sense, the sampling process performed through the Dirac comb [4] isolates each data point, rendering them independent from each other. The function and its derivatives are no longer continuous. By re-establishing the continuity between sampled points, a method that yields a highly accurate numerical Fourier transform can be devised.

2 Theory

Let $t \in \mathbb{R}$ and $f \in \mathbb{R}$. As usual, $\mathbb{C}$ is the field of complex numbers, $\mathbb{R}$ is the set of real numbers, $\mathbb{N}$ the set of nonnegative integers and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. Let us define two rectangular functions:

$$R(t) = \chi(t - 0^-)\, \chi(-t + T^+) \tag{1}$$

and

$$S(t) = \chi(t)\, \chi(-t + T), \tag{2}$$

where χ is Heaviside's function, and let

$$0^- = \lim_{\varepsilon \to 0} (0 - \varepsilon) \ , \qquad T_\alpha^+ = \lim_{\varepsilon \to 0} (T_\alpha + \varepsilon)$$

Let $g : \mathbf{R} \to (\mathbf{R} \text{ or } \mathbf{C})$ be a continuous function which admits derivatives of any order for all t such that $S(t) \neq 0$. Let us now define the function

$$h(t) = R(t)\, g(t) \tag{3}$$

and adopt the following definition of Fourier transforms

$$\mathcal{F}\{h(t)\} = \int_{\mathbf{R}} h(t)\, e^{-i2\pi f t} dt \tag{4}$$

In virtue of Heaviside's and Dirac's delta functions properties [12, 5], the n^{th} derivative of h with respect to t is

$$h^{(n)}(t) = \chi(t - 0^-)\, \chi(-t + T^+)\, g^{(n)}(t) + D_n(t) \tag{5}$$

*Normand Beaudoin is a post-doctoral fellow at the Department of Computer Science, Middlesex College, University of Western Ontario, London, Canada, N6A 5B7. Phone: (519) 661-2111. Email: normand@csd.uwo.ca

†Steven S. Beauchemin is with the Department of Computer Science, Middlesex College, University of Western Ontario, London, Canada. N6A 5B7. Phone: (519) 661-2111. Email: beau@csd.uwo.ca

in which $D_n(t)$ is defined as

$$D_n(t) = \begin{cases} 0 \quad \text{if} \quad n=0 \\ \sum_{m=0}^{n-1} \{g^{(m)}(0^-)\delta^{n-m-1}(t-0^-) \\ -g^{(m)}(T^+)\delta^{n-m-1}(t-T^+)\} \quad \text{if} \quad n \in \mathbf{N}^* \end{cases} \tag{6}$$

Eq. (5) and (6) express the fact that the n^{th} derivative of h with respect to t is the ordinary n^{th} derivative of the function h strictly inside the rectangular box where it is continuous and differentiable, in addition to the n^{th} derivative of h in the region where it is discountinuous.

According to our definition of the Fourier transform, we have:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \int_{-\infty}^{\infty} h^{(n)}(t)\, e^{-i2\pi ft}dt \tag{7}$$

We can expand the integral in (7) into parts to form:

$$\begin{aligned} \mathcal{F}\left\{h^{(n)}(t)\right\} &= \int_{-\infty}^{0} h^{(n)}(t)\, e^{-i2\pi ft}dt \\ &+ \int_{0}^{T} h^{(n)}(t)\, e^{-i2\pi ft}dt + \int_{T}^{\infty} h^{(n)}(t)\, e^{-i2\pi ft}dt \end{aligned} \tag{8}$$

The sum of the first and last integrals of the right handside of (8) is clearly $\mathcal{F}\{D_n(t)\}$. Hence, (8) becomes:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \int_{0}^{T} h^{(n)}(t)\, e^{-i2\pi ft}dt + \mathcal{F}\{D_n(t)\} \tag{9}$$

By separating the interval $[0,T]$ into N equal $\Delta t = T/N$ subintervals, (9) can be rewritten as:

$$\begin{aligned} \mathcal{F}\left\{h^{(n)}(t)\right\} &= \textstyle\sum_{j=0}^{N-1} \left\{ \int_{j\Delta t}^{(j+1)\Delta t} h^{(n)}(t)\, e^{-i2\pi ft}dt \right\} \\ &+ \mathcal{F}\{D_n(t)\}\ , \quad j \in \mathbf{N} \end{aligned} \tag{10}$$

Since $h^{(n)}$, between and at 0 and T is continuous and differentiable, it can be approximated, for $t \in [j\Delta t, (j+1)\Delta t]$, for each $j \in [0, N-1]$, by a Taylor expansion:

$$h^{(n)}(t) = \sum_{p=0}^{\infty} \frac{h_j^{(p+n)}(t-j\Delta t)^p}{p!}\ , \quad p \in \mathbf{N} \tag{11}$$

where $h_j^{(m)}$ is the m^{th} derivative of h at the point $t = j\Delta t$. Merging (10) and (11), using the substitution $\tau = t - j\Delta t$ and performing the adequate permutation of the integral and sums on j and p, we obtain:

$$\begin{aligned} &\mathcal{F}\left\{h^{(n)}(t)\right\} = \\ &\sum_{p=0}^{\infty} \left\{ \left(\int_0^{\Delta t} \frac{\tau^p e^{-i2\pi f\tau}}{p!} d\tau \right) \left(\textstyle\sum_{j=0}^{N-1} h_j^{(p+n)} e^{-i2\pi fj\Delta t} \right) \right\} \\ &+ \mathcal{F}\{D_n(t)\} \end{aligned} \tag{12}$$

To numerically compute the Fourier transform of h, we must evaluate it for some discrete values of f. Let $f = k\Delta f = k/T$, $k \in \mathbf{N}$ be these discrete variables. In addition, let us define H_k as the discrete version of $\mathcal{F}\{h^{(n)}(t)\}$. The integral in (12) depends only on the variable f (or k) and on parameters p and Δt and can be evaluated analytically, whether f is continuous or discrete, once and for all, for each value of p as:

$$I_p = \frac{1}{p!} \int_0^{\Delta t} \tau^p e^{-i2\pi f\tau} d\tau \tag{13}$$

Since the integral in the definition of I_p is always finite and, in the context of Gamma function [9], $p! = \pm\infty$ when p is a negative integer then $I_p = 0$ for $p < 0$.

The summation on j in (12), when $f = k\Delta f = k/T$ is the discrete Fourier transform of the sequence $h_j^{(p+n)}$, $j \in [0, N-1] \subset \mathbf{N}$ [4]. We denote it as $F_{p+n,k}$. Since $\Delta t = T/N$ and $f = k/T$ we have:

$$F_{p+n,k} = \sum_{j=0}^{N-1} h_j^{(p+n)} e^{-i2\pi \frac{kj}{N}} \tag{14}$$

Substituting (13) and (14) in (12), we obtain the following result:

$$\mathcal{F}\left\{h^{(n)}(t)\right\} = \sum_{p=0}^{\infty} I_p F_{p+n,k} + \mathcal{F}\{D_n(t)\} \tag{15}$$

When $n = 0$, (15) becomes:

$$H_k = \sum_{p=0}^{\infty} I_p F_{p,k} \tag{16}$$

Now, integrating by parts the right hand side of (7) yields:

$$\mathcal{F}\left\{h^{(n+1)}\right\} = i2\pi f \mathcal{F}\left\{h^{(n)}\right\} \tag{17}$$

Defining $b_n = i2\pi f\mathcal{F}\{D_n\} - \mathcal{F}\{D_{n+1}\}$, combining (15) and (17) and reorganizing the terms yields:

$$-i2\pi f I_0 F_{n,k} + \sum_{p=1}^{\infty} \left(I_{(p-1)} - i2\pi f I_p\right) F_{p+n,k} = b_n \tag{18}$$

With the following definition:

$$J_\alpha = I_{\alpha-1} - i2\pi f I_\alpha \tag{19}$$

(18) becomes:

$$J_0 F_{n,k} + \sum_{p=1}^{\infty} J_p F_{p+n,k} = b_n \tag{20}$$

Given the definition of g and h, we have $g^{(n)}(0^-) = g^{(n)}(0) = h^{(n)}(0)$ and $g^{(n)}(T^+) = g^{(n)}(T) = h^{(n)}(T)$. Using these facts in addition to the properties of Fourier transforms and those of Dirac delta functions [5], expanding b_n results in the simple following form:

$$b_n = h^{(n)}(T)\, e^{-i2\pi fT} - h^{(n)}(0) \tag{21}$$

In the discrete context, where $f = k/T$, (21) takes a simple form:

$$b_n = h^{(n)}(T) - h^{(n)}(0) = h_N^{(n)} - h_0^{(n)} \tag{22}$$

Up to this point, all equations are rigorously exact since p tends towards infinity. However, in practical situations we introduce approximations by limiting the range on p. Let us define $\theta \in \mathbf{N}$, the truncating parameter. We refer to it as the order of the system. Let us expand (20) for each value of $n \in [0, \theta - 1] \subset \mathbf{N}$. This generates a system of θ equations, and for each of these we let p range from 1 to $\theta - n$. Thereafter, the terms are reorganized to obtain the following system, which is written in a matrix form:

$$\begin{bmatrix} J_1 & \cdots & J_\theta \\ J_0 & \cdots & J_{\theta-1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_1 \end{bmatrix} \begin{bmatrix} F_{1,k} \\ F_{2,k} \\ \vdots \\ F_{\theta,k} \end{bmatrix} \simeq \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{\theta-1} \end{bmatrix} + \begin{bmatrix} -J_0 F_{0,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{23}$$

or more compactly as:

$$\mathsf{M}_b \mathsf{F}_b \simeq \mathsf{B} + \mathsf{C} \tag{24}$$

The general expression for elements of M_b is:

$$(\mathsf{M}_b)_{\mu\nu} = J_{\nu-\mu+1} \tag{25}$$

Let us now write (24) as:

$$\mathsf{F}_b \simeq \mathsf{M}_b^{-1}(\mathsf{B} + \mathsf{C}) \tag{26}$$

Provided a numerically known matrix B (see next section on boundaries conditions), the values of F_b are then obtained from (26). Hence, the terms in (16), for $p \in [0, \theta]$, are completly determined, and the truncated version of (16) can be written as:

$$H_k \simeq \sum_{p=0}^{\theta} I_p F_{p,k} \tag{27}$$

Let us define a one row matrix $\mathsf{I}_\theta = [I_1\ I_2\ \cdots\ I_\theta]$, and write (27) as follows:

$$H_k \simeq I_0 F_{0,k} + \mathsf{I}_\theta \mathsf{F}_b \tag{28}$$

With (28), we approximate the Fourier transform (or the inverse Fourier transform) of a digitized function in one dimension. The digitized Fourier transform calculated with (28) is not band limited (as with the DFT which is periodical). Eq. (28) remains valid and accurate as an approximation of the analytical Fourier transform for all positive or negative values of k (or f) [2].

Eq. (28) contains the symbolic form of F_b which can be used as is to form a single equation without having to numerically evaluate each term of F_b separately. On the other hand, if, for instance, (26) is used to numerically compute each term of F_b for values of k from 0 to $N-1$, it produces θ different sequences of numbers which are approximations of the DFT of the derivatives $h_j^{(p)}$, for values of $p \in [1, \theta]$. Thus, applying the inverse DFT operation to each of these sequences generates the corresponding sequences $h_j^{(p)}$ that are the numerical derivatives of order 1 to θ of the initial function $h_j^{(0)}$. This implies that one can numerically compute the derivatives of any order of a digitized function or signal [2].

Derivatives calculated in that manner are continuous inbetween and at each data point. We obtain spline polynomials of any odd degree, with their corresponding properties, merely with the use of a classical DFT (FFT) [2]. Such high-order spline interpolation polynomials allows integrals between any limit to be accurately computed [2].

3 Boundary conditions

Eq. (26) requires the values of B to be properly computed. Eq. (22) shows that the values of B are directly related to the values of the derivatives of different orders at both ends (0 and T) of the function. The computation of such derivatives based on a few data points near the boundaries of the function is known to be a crude and often inaccurate method. In [2] and [3], a method based on all the data points of the function has been devised, resulting in high numerical accuracy. The drawback is that sometimes a limitation on the order θ of the system is implied. Most of the time that limitation is harmless but, in image processing, it may happen that $\theta < 1$. To overcome that inconvenient, a more robust method that still uses the information from all data points of the function has been developed.. This method is based on the fact that the numerical Fourier transform presented in this paper gives the derivatives of any order up to θ of the initial function. Furthermore, as mentioned earlier, these derivatives form spline polynomials of degree θ that interpolate the sampled initial function h_j. Hence it is possible to compute a spline norm [7] that contains the boundary conditions (B) as parameters. It is then possible to adjust these parameters to minimize the spline norm. Limitation on space does not permit to go throught all the mathematical details of the method. However let us briefly show the main operational equations.

The one column vector or the $\theta \times 1$ matrix B that contains the boundaries conditions is given by:

$$\mathsf{B} = \left[S\left\{\overline{\mathbf{E}}_L \mathbf{E}_l^{\mathsf{T}}\right\}\right]_{\theta\times\theta}^{-1}\left[-S\left\{\overline{\mathbf{E}}_L \Lambda_{lj}\right\}\right]_{\theta\times 1} \quad (29)$$

S is the following three-summation operator:

$$S = \sum_{p=0}^{\theta-\varphi}\sum_{P=0}^{\theta-\varphi} \frac{\Delta t^{p+P=1}}{p!P!\,(p+P+1)} \sum_{j=0}^{N-1} \quad \text{, with } \varphi = \frac{\theta+1}{2} \quad (30)$$

In (29) we define the matrix as follows: $\mathbf{E}_l = [E_{l1j} \cdots E_{l\theta j}]^{\mathsf{T}}$ and $\overline{\mathbf{E}}_L = \left[\overline{E}_{L1j} \cdots \overline{E}_{L\theta j}\right]^{\mathsf{T}}$ in which T means the transpose (not conjugate) and the overbar means the complex conjugate. In addition, $\Lambda_{lj} = D^{-1}\left\{-q_{l1} J_0 F_{ok}\right\}_j$ and $E_{lmj} = \left\{q_{lm}\right\}_j$. D^{-1} stands for the inverse discrete Fourier transform. The term q_{lm} represents the element in the l^{th} row, m^{th} column of the matrix M_b^{-1}. All the elements, except Λ that depends on h, depend on the parameters (N, T) of the function and not on the actual values h_j of the function. Hence, they are computed only once.

4 Time complexity

A close examination of (23) and (28) reveals that the computation of only one classical FFT is required. The other terms form a correcting operation to be applied once on each of the N values of the FFT. The time complexity of the entire correcting operation is $O\left(N\right)$ and the time complexity of the FFT is $O\left(N \log N\right)$. Hence, the time complexity of the entire algorithm is $O\left(N \log N\right)$ when θ is kept constant. The time complexity relatively to $\theta$the order of the system, is $O\left(\theta^2\right)$. However, the error ε on computed results decreases exponentially with an increase of the order θ. The method can be applied sequentially to compute d-dimensional Fourier transforms. In the multidimensional case, for each $\alpha \in \{1, 2, \cdots, d\}$ we have $t_\alpha \in [0, T_\alpha\}$. This interval is separated into N_α equal $\Delta t_\alpha = T_\alpha / N_\alpha$ subintervals, and $f_\alpha = k_\alpha \Delta f_\alpha = k_\alpha / T_\alpha$. As with the ordinary DFT (FFT), the order in which the dimensions are treated is irrelevant. The number of times (28) has to be applied to completely compute the d-dimensional Fourier transform is:

$$PQ \quad \text{where} \quad P = \prod_{\alpha=1}^{d} N_\alpha \quad \text{and} \quad Q = \sum_{\beta=1}^{d} \frac{1}{N_\beta} \quad (31)$$

The time complexity is then $O\left(P \log P\right)$. Let us consider $N_\alpha = a_\alpha N$, $\forall\alpha$, a_α as constants. Then it is easy to show that the time complexity is $O\left(N^d \log N\right)$ which is the same as for the DFT in d-dimensions.

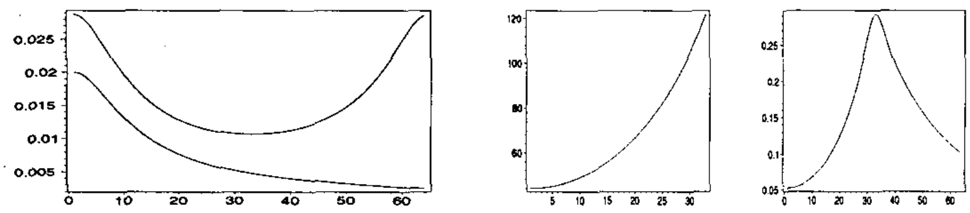
5 Examples in 1 dimension and on image

In this section, an example in one dimension is used to illustrate the algorithm and an image processing example is presented. The one dimensional function is a decaying exponential $(\exp(-50t))$ representing a frequent situation in images where values changes rapidly within a few pixels only. Figures 1 shows, respectively, the classical DFT and the new discrete Fourier transform $(NewDFT)$, both evaluated from the $N = 64$ data points of the exponential decay. The usual periodical behavior of the DFT is evident and the absence of periodicity of the $NewDFT$ is clearly seen. The same figure also shows the percentage of error of the DFT (fairly computed on the first half of the period only) and the error of the $NewDFT$ computed on the full number of data points. The error of the DFT is between 40% and 120% while the error of $NewDFT$ (with $\theta = 9$) is between 0.05% and 0.25%.

Figure 2 shows the image of a simple occlusion scene involving two sine plaids, for which the Fourier transform must be computed to reveal the frequency structure of the occlusion. This type of images is relevant to theoretical and practical studies in image processing [1]. The same figure also shows the DFT and the $NewDFT$ ($\theta = 3$) of the sine plaid occlusion scene. It should be noted that the initial signal is 64x64 and the transformed signals have been computed on 128x128 data points centred on the origin $(f_1, f_2) = (0, 0)$. One can clearly see the repeated lines and peaks due to periodicity of the classical DFT. On the other hand, the $NewDFT$ completely eliminates that annoying behavior. Even on an extended number of data points, the $NewDFT$ gives an excellent approximation of what would be the analytical Fourier transform (often difficult if not impossible to compute) of the image.

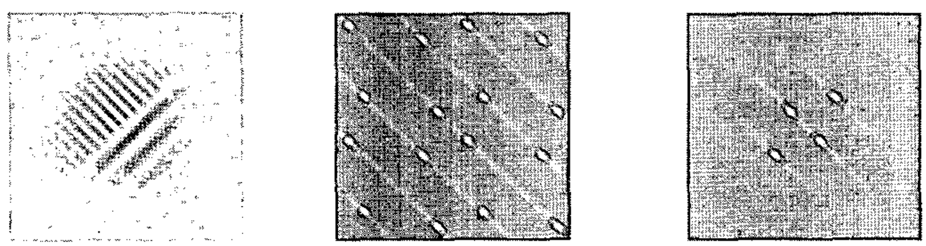
6 Conclusion

The method presented in this contribution provides accurate approximations of the continuous Fourier transform, is no longer periodical and yields a broader frequency domain than the usual half-period of the DFT. The method gives accurate numerical partial derivatives of any order and the polynonial splines of any odd degree with their optimal boundary conditions. The time complexity is the same as for the FFT. The time complexity, relatively to θ (independent of the time complexity related to N) is $O\left(\theta^2\right)$ while the accuracy increase exponentially with θ. Hence, the numerical accuracy increases much more rapidly than the computational cost of the proposed method. Finally, results show that a significant improvment is accomplished in image processing throught the complete elimination of periodicity in the computed numerical Fourier transform.



DFT (top) and $NewDFT$ (bottom) Error on DFT Error on $NewDFT$

Figure 1. Classical discrete Fourier transform (DFT, top curve) and discrete Fourier transfom given by the new method ($NewDFT$, bottom curve) and percentage of error on DFT computed on the first half of data points only and percentage of error on $NewDFT$ computed on the full range of data points. The exact Fourier transform is not shown since it is indistinguishable from $NewDFT$.



Occlusion DFT of occlusion $NewDFT$ of occlusion

Figure 2. Occlusion of a two dimensional signal by an another of different frequency and orientation and the discrete Fourier transform of the occlusion computed with the classical DFT and with the $NewDFT$.

References

[1] S. Beauchemin and J. Barron. On the fourier properties of discontinuous motion. *Journal of Mathematical Imaging and Vision*, 13:155–172, 2000.

[2] N. Beaudoin. A high-accuracy mathematical and numerical method for fourier transform, integral, derivative, and polynomial splines of any order. *Canadian Journal of Physics*, 76(9):659–677, 1998.

[3] N. Beaudoin and S. Beauchemin. Accurate numerical fourier transform in d-dimensions. *TR-561, Dept. Computer Science, The University of Western Ontario*, pages 1–14, 2001.

[4] E. O. Brigham. *The fast Fourier transform*. Prentice-Hall, 1974.

[5] E. Butkov. *Mathematical Physics*. Addison-Wesley, Reading, 1968.

[6] M. Froeyen and L. Hellemans. Improved algorithm for the discrete fourier transform. *Rev. Sci. Instrum.*, 56(12):2325, 1985.

[7] K. Ito. *Encyclopedic Dictionary of Mathematics*. Mathematical Society of Japan, The MIT Press, 1993.

[8] C. Kittel. *Physique de l'état solide*. Dunod Université, 1983.

[9] E. Kreyszig. *Advanced engineering mathematics*. Wiley, 1983.

[10] S. Makinen. New algorithm for the calculation of the fourier transform of discrete signals. *Rev. Sci. Instrum.*, 53(5):627, 1982.

[11] L. Marchildon. *Mécanique Quantique*. De Boeck Université, 2000.

[12] N. Morrison. *Introduction to Fourier analysis*. Wiley-Interscience, 1994.

[13] J. Schutte. New fast fourier transform algorithm for linear system analysis applied in molecular beam relaxation spectroscopy. *Rev. Sci. Instrum.*, 52(3):400, 1981.

[14] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[15] S. Sorella and S. K. Ghosh. Improved method for the discrete fast fourier transform. *Rev. Sci. Instrum.*, 55(8):1348, 1984.