Fast and accurate computation of the Fourier transform of an image

Gregory Beylkin

University of Colorado at Boulder, Program in Applied Mathematics Boulder, CO 8O3O9O526

ABSTRACT

We use the Battle.Lemarié scaling function in an algorithm for fast computation of the Fourier transform of a piecewise smooth function f. Namely, we compute for —N  m, n N (with a given accuracy) the integrals

* ,1 1 • f(m, n) = I I 1(2;,y) dy (0.1) Jo JO
in O(ND)+O(N2 log N) operations, where ND is the number of subdomains where the function f is smooth. We consider an application of this algorithm to image processing. Notwithstanding that it might be advantageous to consider an image as a piecewise smooth function f, it is a common practice in image processing to simply take the FFT of the pixel values of the image in order to evaluate the Fourier transform. Due to the jump discontinuities of the function f, the accuracy of such a computation is poor.
We propose our algorithm as a tool for the accurate computation of the Fourier transform of an image since the direct evaluation of (0.1) is very costly.

1. IMAGE AS A PIECEWISE SMOOTH FUNCTION

It is natural and useful to consider an image as a piecewise smooth function. For example, the goal of segmentation algorithms is to find the boundaries of smooth subdomains of an image. Furthermore, at the level of pixels, one may consider an image as a collection of tiny squares with different values so that the total image is a linear combination of characteristic functions of elementary squares as in the example in Figure 3.
Yet, computing the Fourier transform of a piecewise smooth function is not an entirely trivial matter, especially if the number of discontinuities of f is large or the subdomains are complicated. If we use the FFT of the pixel values of an N x N image (which is equivalent to using the trapezoidal rule in (0.1)) then, in some directions, the error will decay only as 1/N. In other words, instead of a piecewise smooth function, we work with an oscillatory function as in the example in Figure 1.
The cost of the direct evaluation of (0.1) for images is prohibitive. The direct algorithm for evaluating the Fourier transform of a linear combination of characteristic functions of elementary squares (or rectangles)

computes f(m, n) = E fj(m, n), for —N  m, n  N, as a sum of contributions from each rectangle
[al, b,] x [ci, d1],

e2mb1 —e21m(1 e2'1' — e2T'1

fz(m, n) = zi (—2irim) (—2irin)

(1.2)

where zi are constants, 1 = 1,..., ND and ND is the number of rectangles. Though accurate, such evaluation of the Fourier transform requires O(N2 .ND) operations and since typically for images ND N2, the direct approach is not practical. Thus, there is a need for a fast and accurate algorithm to evaluate (0.1).

244 ISPIE Vol. 2277 0819446010/94/$6.OO

f8\m+l ir)

1.

Figure 1: By taking the Fourier transform of pixel values we effectively replace the characteristic function of an interval by an oscillatory function with the same pixel values.

2. FAST ALGORITHM

The problem of computing (0.1) was first considered in [7] where it was motivated by the needs of VLSI design. The algorithm of [7] uses Gauss-Legendre quadratures to evaluate portions of (0.1) and Lagrange interpolation to redistribute the resulting values to an equally spaced grid. The result is then obtained by using the FFT.
In [2] the problem of fast computing of (0.1) is addressed by projecting f on a subspace of a Mul-tiresolution Analysis (MBA) effectively bandlimiting the function f. The algorithm appears to be more efficient than that in [7]. In fact [2] considers a more general problem of computing the Fourier transform of generalized functions with singularities of the type 4/r(A + 1), where r is the gamma function and x4 is defined as zA for x > 0 and zero for x

and consider the periodic function a(m),

l=oo 1=m

a(m) = E I(m) (+1) 12 : (2m+1)(1) e2nuZ. (2.5)

1=-rn

where (2rn+1)(1) are values of the central B-.spline of order 2m + 1. We also need the Fourier transform of the Battle-Lemarié scaling function [5], [1],

(rn) ((rn)() =

' I
• (2.6)
yIa(rn)()
Our approach is based the following

Theorem 1 Let E be the error of the_approximation of the Fourier transform Iofthe generalized function I by the periodic function 2i/2F()//Jm)(),

E = sup 123/2 F() J(2i)1 / p 1f(2')I, i < 0, (2.7) II° /a()()

where a is a parameter, a(m) is given in (2.5) and F is the Fourier series,

F() = fke_2T, (2.8)

kEZ

with coefficients fk gwen in (2.3).
1. If

If()I  C(1 + IIY a <rn,
then we have

1 1 m+1 E 2(a) 1 (a) + C1(O, a) l=2... C1(l,a) (a a)] (2.9)

where

C1(l,a) = sup lJ(2'(+l))l. (2.10) IIa
2. For any e> 0 we may choose the order m of the central B-spline and a > 0 so that for I  a

E  e. (2.11)

The proof of Theorem may be found in [2] where it is also shown that

23/2 F() = J(2_i(±l)) (rn)(+l) = J(2'e) (m)(e) + J(2_i(i)) (rn)(e+l) (2.12)

1EZ

As an illustration of why the left hand side of (2.12) is a good approximation of f(2i) for  a, we plot
(rn)(e), (m)(+ 1) and a(rn)(1) for rn = 23 in Figure 2. We note that for a = 1/4 and a the

246 / SPIE Vol. 2277

0.75

0.50 . .. - . ...: • • - . . o: — I I —

1 L—I
1(i) = EIke
Lh/2a(m)(l/L) k=O

-2.OO-1J5-1.5O-1.2S-1.OO-O.7SO.SO-O.2S 0 0.250.500.15 1.00 1.25 1.50 1.752.00

Figure 2: The Fourier transform of Battle-Lemarié scaling function of order m = 23. Shown are functions

(m)(), (m)(+ 1) d (m)(1).

values of (m)() are equal to one (with double precision accuracy), whereas (m)(.... 1) d (m)(1)
are equal to zero with the same precision.
Setting a = 1/4 and using Theorem 1, we have (with accuracy e)

f(i) = : (2.13)

L1/2 f(m) (ilL) k€Z

for —L/4  1  L/4, where L = 2'. We may always arrange fk = 0 for k < 0 and k  L. Thus, we replace the series in (2.13) by a finite sum and obtain

, .-L/4  1  L/4, (2.14)

which may be evaluated using the FFT.
For computing the Fourier transform of an image, we have (setting a = 1/4)

N—i N—i

J(n, n') =

1 : fkk' (2.15)
Na(m)(n/N) a(m)(nu/N) ,=o k'=O

for —N/4 n,n' N=2', and

.fkk' L L) 3(x) /3)(y) dzdy. (2.16)
The acctiracy is controled by an appropriate choice of the order of the spline m.
As a result we have a simple algorithm based on the approximation in Theorem 1 which consists of three steps

SPIE Vol. 2277 I 247

1. Computing integrals (2.16). The cost of generating the matrix fjj is proportional to m2ND. The spline order m is usually chosen proportional to log €, the number of accurate digits.

2. Computing the FFT of the matrix fkrs UI order to compute the sum in (2.15). This step requires O(N2 log N) operations.

N T, TFFT Tm Error E T10 Td,.

64 0.06 0.50 0.02 4.4e-15 0.58 0.47

128 0.17 2.88 0.07 2.4e-15 3.12 1.89

256 0.60 12.25 0.31 1.3e-15 13.16 7.58

512 2.28 54.32 1.23 1.Oe-15 57.83 30.41

Table 1: Timing and accuracy for Example 111.1 on SPARC-b

values instead of computing them, etc., will undoubtedly further improve the performance of the algorithm. These issues will be addresed elsewhere.
In what follows we denote by 1',,, the run time for the projection step of the algorithm where we compute the integrals in (2.16). TFFT denotes the run time for the FFT step, and Tm for the third (multiplication) step of the algorithm. The error E is the maximal absolute error among all computed frequencies obtained by comparing the output of the algorithm with that of the direct evaluation. T0 denotes the total run time, T0 = T + TFF'T + Tm . For comparison TSOR denotes the run time from [7] for a similar experiment. Tdj denotes the run time required for the direct computation.

Example 111.1. We compute the Fourier transform ofthe function f which is a constant and has a rectangle of area 0.64 as its support. We compare the run time of our algorithm with that of the direct evaluation for N = 2's, = 6, 7, 8, 9 and report the results in Table b. We observe that the direct algorithm for a single rectangle is faster only by a factor of 2.

Example 111.2. In this example we consider f to be a linear combination of characteristic functions of a pseudo-random combination of 1225 rectangles of the total area of 0.64 and the perimeter 112. Each rectangle was projected separately. The results are shown in Table 2. For N = 2's, 6, 7 we compare the run time and accuracy with that of the direct algorithm. We note that in this example our algorithm is dominated by the FFT step. The speed up factor compared with the direct evaluation is 500.
We also compare the run times with the algorithm in [7]. The run times of the algorithm in [7] are modified by a factor 0.21 which was obtained by comparing with the run times of the direct evaluation. We observe that the speed up is at least by a factor of bO. Our algorithm also requires 4 times less memory. Comparison with the direct evaluation is for illustration purposes only. '

Example 111.3. In this example I is a linear combination of characteristic functions of a pseudo-random combination of 40, 000 rectangles of the total area of 0.64 and the perimeter 640. The results are shown in Table 3. As individual rectangles become smaller, we start to observe that the first step of the algorithm practically does not depend on the number of frequencies (i.e., size of the matrix).

Example 111.4. In this example I is a linear combination of characteristic functions of a pseudo-random combination of b60, 000 rectangles of the total area of 0.64 and the perimeter b280. The results are shown in Table 4. Again we observe that T practically does not change as N changes from 64 to 5b2.

'In [7] integration by parts is used to reduce the integral over the domain to that over the boundary. A similar approach may be taken here as well with some reduction in the number of operations in the projection step of the algorithm.

SPIE Vol. 2277 / 249

—f---1',TFFTTmError ET101TSOR *0.21Td,r
2.000.500.024.Oe-152.5111.55581.75
2.462.850.072.2e455.3844.732,331.8
3.3012.470.3116.08210.849,324 (est.)
6.1753.841.2461.25NA37,252 (est.)
Table 2: Timing and accuracy for Example 111.2
—f-—T,TFFTTmTj(sec)Estimate of T.uinhours
.___i_60.760.490.0261.275.2 h
62.452.870.0765.3921 h
64.8812.230.3177.4284.2 h
72.1954.331.26127.78337.9h
Table 3: Timing for Example 111.3 (40,000 rectangles)
TTFFTTmT01 (SEC)Estimate of Td1 in hours
.__±_242.580.510.02243.1120.9h
.._!a248.572.920.08251.5784h
250.8012.530.32263.65336.9 h—--261.5255.931.22318.671,351.6 h
Table 4: Timing for Example 111.4 (160,000 rectangles)
250 1 SPIE Vol.2277

SPIE

Figure 3: Letters SPIE as an image constructed from elementary squares.

[3] C. Chui. An Introduction to Wavelets. Academic Press, 1992.
[4] A. Dutt and V. Rokhlin. Fast Fourier Transform for Nonequispaced Data. SIAM J. Sci. Stat. Comp., 1993. to appear.
[5] P.G. Lemarié. Ondelettes a localisation exponentielles. J. Math. Pures et Appi., 67:227—236, 1988.

[6] I.J. Schoenberg. Cardinal Spline Interpolation. SIAM, 1973. CBMS-NSF Series in Applied Math. #12. [7] E. Sorets. Fast Fourier Transform of Piecewise Constant Functions. 1993. Yale University Research Report, YALEU/DCS/RR-986.
[8] J. 0. Stromberg. A Modified Franklin System and Higher-Order Spline Systems on It" as Unconditional Bases for Hardy Spaces. In Conference in harmonic analysis in honor of Antoni Zygrnund, Wadworth
math. series, pages 475—493, 1983.
[9] M. Unser and A. Aidroubi. Polynomial splines and wavelets - a signal processing perspective. In C. Chui, editor, Wavelets: A Tutorial in Theory and Applications, pages 91—122. Academic Press,
1992.

252 / SPIE Vol. 2277