Quasi fast Hankel transform

A. E. Siegman

Edward L. Ginzton Laboratory and Department of Electrical Engineering, Stanford University, Stanford, California 94305

Received March 25, 1977

We outline here a new algorithm for evaluating Hankel (Fourier–Bessel) transforms numerically with enhanced speed, accuracy, and efficiency. A nonlinear change of variables is used to convert the one-sided Hankel transform integral into a two-sided cross-correlation integral. This correlation integral is then evaluated on a discrete sampled basis using fast Fourier transforms. The new algorithm offers advantages in speed and substantial advantages in storage requirements over conventional methods for evaluating Hankel transforms with large numbers of points.

In many optical calculations one would like to evaluate Hankel transforms numerically with computational efficiency analogous to the fast Fourier transform. We describe here a new "fast Hankel transform" (FHT) algorithm that can evaluate Hankel transforms while providing many of the desired benefits.

The standard Hankel transform of order l is given by

$$g(\rho) = 2\pi \int_0^\infty rf(r)J_l(2\pi\rho r)\,\mathrm{d}r \tag{1}$$

with a symmetric reverse transform integral from $g(\rho)$ to $f(r)$. Using the change of variables $r = r_0 e^{\alpha x}$, $\rho = \rho_0 e^{\alpha y}$, where r_0, ρ_0, and α are initially arbitrary, converts the one-sided Hankel transform integral [Eq. (1)] into the two-sided cross-correlation integral

$$\hat{g}(y) = \int_{-\infty}^{\infty} \hat{f}(x)\,\hat{j}(x+y)\,\mathrm{d}x, \tag{2}$$

in which the input and transform functions are $\hat{f}(x) = rf(r)$, $\hat{g}(y) = \rho g(\rho)$, and the modified Bessel kernel is $\hat{j}(x + y) = 2\pi\alpha r\rho\, J_l(2\pi r\rho)$. This change of variables might be called the "Gardner transform." [1] It has been applied elsewhere to exponential or Laplace transforms[2–4] and can potentially be applied to any integral transform whose kernel has a product argument.[5]

A correlation integral such as Eq. 2 can be evaluated efficiently in a discrete sampled approximation by using Fourier-transform methods. Sampling the functions $\hat{f}(x)$ and $\hat{g}(y)$ at discrete values given by $x = n$, $y = m$ with $n, m = 0,1,\ldots,N-1$ is equivalent to sampling the original functions $f(r)$ and $g(\rho)$ at discrete exponentially increasing values given by

$$r_n = r_0 e^{\alpha n}, \qquad \rho_m = \rho_0 e^{\alpha m}. \tag{3}$$

These sampled values are confined to truncated ranges $r_0 \leq r < b$ and $\rho_0 \leq \rho < \beta$ with upper limits $b = r_0 e^{\alpha N}$ and $\beta = \rho_0 e^{\alpha N}$. The product βb is a space–bandwidth product for the transform. In terms of the sampled values $f_n = \hat{f}(x_n) = r_n f(r_n)$, $g_m = \hat{g}(y_m) = \rho_m g(\rho_m)$, and $j_{n+m} = \hat{j}(x_n + y_m) = 2\pi\alpha r_n \rho_m J_l(2\pi r_n \rho_m)$, the Hankel transform integral [Eq. (1) or Eq. (2)] is approximated to some level of accuracy by the discrete sum

$$g_m = \sum_{n=0}^{N-1} f_n j_{n+m}. \tag{4}$$

If the input sequence f_n is padded with zeros and the upper limit of the sum extended to $2N - 1$, this sum becomes a $2N$-term discrete circular correlation. This discrete correlation can be evaluated exactly using the discrete $2N$-term fast Fourier transforms of the input sequences f_n and j_n by carrying out the steps

$$g_m = \mathrm{FFT}\,[\mathrm{FFT}(f_m) \times \mathrm{FFT}^*(j_m)]. \tag{5}$$

The FFT and FFT* notations indicate the forward and inverse ($\pm j$) Fourier transforms (or vice versa). The two FFT's inside the brackets are multiplied together term by term before performing the outer FFT.

To the extent that the values g_m generated by Eq. (4) or Eq. (5) represent a good approximation to the true sampled transform values $\hat{g}(y_m)$, this procedure computes an N-point discrete approximation to a Hankel transform by computing two $2N$-term fast Fourier transforms plus $2N$ additional multiplications. The transforms are performed in place, requiring only the additional storage of $2N$ previously computed values of the Fourier transform of j_{n+m}. The extra values of g_m from $m = N$ to $m = 2N - 1$ represent aliased results from the circular correlation process and are discarded.

Criteria for choosing the parameters r_0, ρ_0, and α can be developed as follows. If β is the largest value of ρ in the transform domain, the highest frequency component in the r domain is $J_l(2\pi\beta r) \sim \cos(2\pi\beta \mathrm{r})$. We then require that the lower-end truncation point in the r domain not be larger than $1/K_1$ cycles of the highest spatial frequency β, or $r_0 = (K_1\beta)^{-1}$. Similarly we require that the sample point spacing $\Delta r_n \equiv r_{n+1} - r_n$ at the upper truncation point $r = b$ not be greater than $1/K_2$ cycles of β, or $\Delta r_N \approx \alpha b = (K_2\beta)^{-1}$. Combining these arguments with exactly analogous arguments in the ρ domain, namely $\rho_0 = (K_1 b)^{-1}$ and $\Delta\rho_N = \alpha\beta = (K_2 b)^{-1}$, leads to the set of relations

$$\begin{gathered} N = K_2\beta b \ln(K_1\beta b), \\ \alpha e^{\alpha N} = K_1/K_2, \\ r_0\rho_0 = (K_2/K_1{}^2)\alpha. \end{gathered} \tag{6}$$

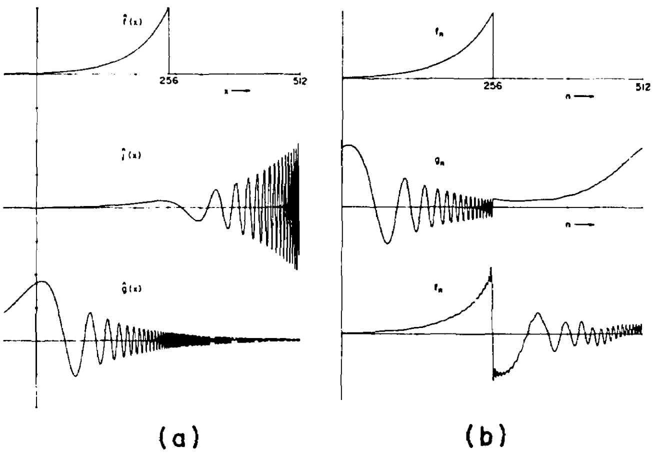

Fig. 1 (a) Exact analytical functions $\hat{f}(x)$, $\hat{j}(x)$, and $\hat{g}(x)$ corresponding to a uniform circular "top-hat" input function $f(r)$ and its Airy-disk Hankel transform, including the top-hat input function $\hat{f}$ (top curve); the $l = 0$ Bessel-function kernel $\hat{j}$ (middle curve); and the Airy-disk Hankel transform $\hat{g}$ (bottom curve). Parameter values for the change of variables are $\alpha = 0.01612$, $r_0 = \rho_0 = 0.06349$, and $\beta = b = 3.938$, corresponding to $N = 256$, $K_1 = K_2 = 4$. The "dimples" in the $\hat{g}$ function at larger arguments are spurious results of a plotting routine with inadequate point spacing. (b) Discrete Hankel transformation and then backtransformation of a sampled top-hat input sequence f_n (top curve) into the transformed sequence g_n (middle curve) and then back into a sequence f_n (bottom curve), using the FHT algorithm with $N = 256$, $K_1 = K_2 = 4$. The portions of the transformed and backtransformed sequences g_n and f_n above $n = 256$ arise from aliasing effects in the transform algorithm and are discarded. The backtransformed sequence f_n exhibits Gibbs phenomena because of the finite truncation of the g_n sequence before it is backtransformed.

We hypothesize that the "points-per-cycle" parameters K_1 and K_2 should both be ≥ 2 points per cycle. Equations (6) then determine α, $r_0\rho_0$, and βb appropriate to a given number of points N; or alternatively the number of points N and the α and $r_0\rho_0$ parameters required to transform with adequate accuracy a function–transform pair having a given space–bandwidth product βb.

This algorithm was programmed in FORTRAN using single-precision arithmetic on an IBM 370/168 computer. Figure 1(a) shows for purposes of comparison analytical plots of a "top-hat" input function $f(r) = 1$, $0 \leq r \leq b$, transformed into $\hat{f}(x)$; the modified Bessel kernel $\hat{j}(x)$; and the Airy-disk function $\hat{g}(x)$ that is the exact Hankel transform of a uniform input; all plotted as continuous functions for parameter values corresponding to $l = 0$, $N = 256$, and $K_1 = K_2 = 4$ points/cycle. The kernel function $\hat{j}(x)$ is seen to be slowly divergent at large x.

Figure 1(b) shows the discrete numerical results generated by the FHT algorithm for this case. The FHT algorithm samples the input function $\hat{f}(x)$ [or alternatively $\hat{g}(y)$] at integer values from 0 to 255, pads the input sequence with zeros from 256 to 511, performs a discrete circular convolution against 512 sampled points from the kernel $\hat{j}(x)$, and yields an output sequence that approximates the exact Hankel transform over the range 0–255 and contains aliased results over the range 256–511.

In Fig. 1(b) the middle curve shows the output sequence g_m generated by the FHT algorithm for the "top-hat" input sequence of the top curve, using $N = 256$ and $K_1 = K_2 = 4$. The Airy disk transform $g(\rho) = 2J_l(2\pi b\rho)/\rho$ is accurately reproduced by the FHT algorithm for $0 \leq m < N$, with irrelevant aliased values for $m \geq N$. The sequence g_m, when truncated and fed back through the FHT algorithm, provides a fair reproduction of the original top hat, as shown in the bottom curve, but with obvious Gibbs-phenomena ripples resulting from the finite truncation of the Airy-disk transform.

The Laguerre–Gaussian functions $u_{pl}(r) = (2\pi r)^2 L_p{}^l(2\pi r^2) \exp(-\pi r^2)$ are self-transforming under Hankel transformation. Figure 2 shows in r,ρ coordinates the FHT algorithm applied twice in succession to a Laguerre–Gaussian function with $l = 0$ and $p = 8$, using $N = 128$ with $K_1 = K_2 = 2$. The mean-square error on successive transforms in this case is $\lesssim 0.4\%$. Laguerre–Gaussians have been accurately transformed out to $p \approx 100$ for $N = 1024$, $K_1 = 8$, and $K_2 = 2$.

Figure 3 shows a top hat and also an annular function transformed and then backtransformed using the FHT algorithm. The observed Gibbs-phenomena ripples are expected for finite truncation in the transform domain whatever Hankel transform algorithm may be employed. They can presumably be reduced by using proper windowing (apodizing). The "droop" in the function values at low r is understood and can be eliminated also.

A small number of experiments indicate that the FHT algorithm works equally well for higher-azimuthal-order Hankel transforms. Beyond the preliminary tests summarized here, the capabilities of this

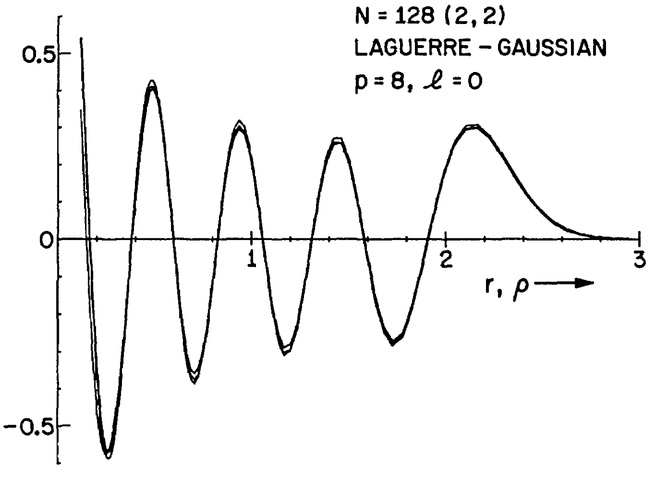

Fig. 2 The input function and two successive FHT's of a Laguerre–Gaussian input function with $p = 8$, $l = 0$ using the FHT algorithm with $N = 128$, $K_1 = K_2 = 2$. The first FHT lies essentially on top of the input function at low arguments and then drops slightly lower. The second FHT is initially slightly lower, then follows closely the first FHT. The curves are plotted as straight lines between the discrete sample points; the discrete point spacing is evident at larger arguments.

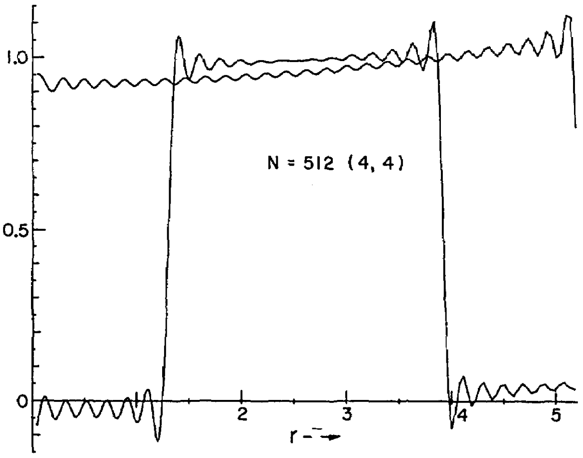

Fig. 3 Results of Hankel-transforming and then back-transforming a top-hat input function $f(r) = 1, 0 \leq r \leq b$, and a uniform annular function $f(r) = 1, b/4 \leq r \leq 3b/4$, using the FHT algorithm with $N = 512, K_1 = K_2 = 4$ ($\alpha = 0.0091648, r_0 = 0.0478665, b = 5.223$). The slump in the top-hat response and, to a lesser extent, in the annular response at low r is related to the truncation at ρ_0 of the narrow Airy-disk Hankel transform in the ρ domain; rescaling of the r and ρ coordinates to give comparable widths in the two domains gives much better results.

algorithm remain to be fully determined (and optimized). One can compare the computational efficiency of the FHT algorithm with straightforward numerical evaluation of Hankel transforms using trapezoidal integration at uniformly spaced sample points with K points per cycle in both r and ρ domains. Assuming that the Bessel function values will be stored in a lookup table, this calculation will require $N = K\beta b$ sample points in each domain and $\frac{1}{2}N^2 = \frac{1}{2}(K\beta b)^2$ complex multiplications per transform. The FFT algorithm, including the aliased values, requires a larger number $2N = 2K_2\beta b \ln(K_1\beta b)$ of sample points because of the unavoidable oversampling with exponentially spaced sample points. However, only $\sim 2N \log_2 4N$ complex multiplications are required to compute the FFT's plus additional steps in the FHT algorithm. The FHT algorithm requires significantly fewer operations for space–bandwidth products above $\beta b \approx 140$ for $K = K_1 = K_2 = 2$ or above $\beta b \approx 70$ for $K = K_1 = K_2 = 4$ points per cycle. One might also hope to obtain improved accuracy against round-off errors because of the superior properties of the FFT algorithm, although this point has yet to be tested.

Of greatest practical importance, however, is the reduction in computer storage requirements using the FHT algorithm. Straightforward Hankel-transform evaluation using stored Bessel-function values requires storing $\frac{1}{2}(K\beta b)^2$ real values of the Bessel kernel $J_l(2\pi r\rho)$ for equally spaced values of both r and ρ (or else the use of interpolation techniques that require additional multiplications). The FHT algorithm, in contrast, is carried out in place, requiring storing only $8K_2\beta b \ln(K_1\beta b)$ total real numbers in the function and kernel arrays. For $\beta b = 1000$ and $K_1 = K_2 = 2$, this would require ~500 Kbytes of storage, assuming 4 bytes per real number, as against ~8000 Kbytes in the standard case.

These advantages in speed, storage requirements, and potential accuracy should make the FHT algorithm valuable in optical-beam and resonator calculations as well as in other optical and nonoptical applications. Further tests and improvements on the basic algorithm will be reported in future publications.

This work was carried out with support from ERDA through the Los Alamos Scientific Laboratory; from United Technology, Pratt & Whitney Aircraft; and from the Air Force Office of Scientific Research.

References

1. D. G. Gardner, J. C. Gardner, G. Lausch, and W. W. Meinke, "Method for the analysis of multi-component exponential decays," J. Chem. Phys. **31**, 987 (1959).
2. J. Schlesinger, "Fit to experimental data with exponential functions using the fast fourier transform," Nucl. Instrum. Methods **106**, 503 (1973).
3. M. R. Smith and S. Cohn-Sfetcu, "Comments on fit to experimental data with exponential functions using the fast fourier transform," Nucl. Instrum. Methods **114**, 171 (1974).
4. S. Cohn-Sfetcu, M. R. Smith, S. T. Nichols, and P. L. Henry, "A digital technique for analysing a class of multicomponent signals," Proc. IEEE **63**, 1460 (1975).
5. S. Cohn-Sfetcu, M. R. Smith, and S. T. Nichols, "On the representation of signals by basis kernels with product argument," Proc. IEEE **63**, 326 (1975).