International Journal of Education and Science Research Review
E-ISSN 2348-6457
Volume-3, Issue-1
February- 2016
www.ijesrr.org
Email- editor@ijesrr.org

A Fast Hankel Transform Methodology and Architecture

Amitava Ghosh
The authors are with Sankalp Semiconductor Pvt. Ltd

Anindya .S. Dhar -Tanmai Kulshreshtha
Department of Electronics and Electrical Communication Engineering
IIT Kharagpur

ABSTRACT:
In this paper, a fast Hankel transform methodology and architecture based on a filtered back-projection method is provided. Hankel transform is required in many applications and if it is computationally intensive, a dedicated hardware and hence architecture is required (especially in real time applications). The Hankel transform is broken into a Fourier transform, followed by an integration (back projection) operation. The Fourier transform can be solved by a fast Fourier transform block requiring $O(N\log_2 N)$ computation time. However, the integration operation takes $O(N^2)$ complexity. In the latter, the delay is due to complex exponential multiplications. This can be solved in much reduced time by CORDIC algorithm which is the paper's main subject. It is shown that the error introduced by CORDIC is small compared to the method's inherent error. Also here, since all the components are uniformly scaled by CORDIC, the scale factor compensation is not required. The method has been designed in MATLAB in bit-level and tested with known input functions. The architecture following from the methodology has been explained. Agreement with theoretical values has been obtained. The method has been used in a computed tomography application as a case study. Extension to voxel based reconstructions has also been shown.

INDEX TERMS: Hankel transform, Back-projection method, CORDIC algorithm, Architecture, Computed tomographic system.

I. INTRODUCTION:

HANKEL transforms arise in applications some of which include computed tomography [1], adaptive optics [2], and radar applications [3]. In many of these applications [2,4], the computational burden is extremely high and a dedicated hardware is necessary (especially for real time applications). In this paper, we propose a fast methodology (basically reducing the computation time of a particular section that has the largest time complexity) and architecture for computing Hankel transforms that is based on the back-projection method and can be used to compute integral transforms of any order.The algorithm is based on [5] which breaks the Hankel transform into a Fourier transform, which can be efficiently solved by fast Fourier transform (FFT) block in $O(N\log_2 N)$ time, and an integration (back-projection) operation. The back-projection operation requires $O(N^2)$ computation time (most computationally intensive operation), where N is the number of input samples. Complex exponential multiplications are performed inside the backprojection operation.

One solution is to use multipliers. The multiplier delay is proportional to the number of bits in the word times the word-length-adder delay which is high for computationally intensive systems (e.g. 64-bit system). Array multipliers occupy huge area and so are inefficient for usage. Instead of using multipliers (which has been used in all the previous works), the complex exponential multiplications can be efficiently performed by the coordinate rotation digital computer (CORDIC) method [6]. Here, the angles need not be stored. Also the delay is independent of the bit length, in fact lesser, say 10 times the word-length-adder delay, which makes it faster than the multiplier counterpart. The CORDIC block however introduces more errors in the computed product than multipliers. It is shown that the error introduced by the CORDIC block is much less compared to the inherent error sources of the back-projection method. Hence performance improvement by CORDIC is possible. Another improvement in computation time is possible because unlike in an FFT, all components are multiplied by the CORDIC scale factor and hence scale factor compensation block is not necessary (e.g. image processing applications, all pixels are scaled by the same factor).

The rest of the paper is organized as follows: In Section II literature review of different Hankel transform methods is provided to show that the used method is the fastest. The modification of the methodology to incorporate the CORDIC section and its architecture is presented in Section III together with the error analysis. In Section IV the results when testing with known input function is provided. In Section V, the application of the methodology in reconstructing a 2-d cross-section of a 3-d object with good peak signal to noise ratio (PSNR) figures (>50 dB) at reduced computation time is shown and effect of non-idealities (viz. noisy inputs) analyzed. Extension to voxel based reconstruction is shown. The conclusion is presented in Section VI.

II. LITERATURE REVIEW OF HANKEL TRANSFORMS METHODS:

Direct application of numerical quadrature is computationally intensive and requires storage of a large number of Bessel functions. The exponential change of variables proposed by Siegman [7] is fast, however requires exponential sampling which can give rise to large errors for fast changing functions. The asymptotic method [8] also produces large errors for small values of the transformed variable. Much of the Hankel transform literature is based on the projection theory. Two types of sub-methods are present. One is the forward projection and the other is the back-projection.

Oppenheim et al. have developed a forward projection method [9], however its computation complexity is high of the $O(N^2)$. Mook [10] and Hansen [11] have developed efficient algorithms for forward projection methods, however it is for the zeroth order only in which the Chebyshev transform reduces to the Abel transform. Among the back-projection methods, Candel [12] has made significant headway. His work was extended to higher orders by Higgins and Munson [5]. They further used symmetry conditions to reduce the number of computations by two. Here, the Hankel transform is broken into a FFT filtered by the input variable followed by integration (back-projection) over the angular limits. The FFT takes $O(N\log_2 N)$ time while the integration requires $O(N^2)$ time. Since some of the computation is performed by the FFT, the overall time is reduced. Suter's [13] algorithm produces comparable operation count with Higgins and Munson's [5], however, it is forward projection based method (Chebyshev transform followed by a Fourier transform). Other recent methods [14-15] have focused on zeroth order transform. Knockaert [16] has given $O(N\log_2 N)$ algorithm for zeroth order transforms via Mellin's approach. The work in [17] finding the Hankel transform using Haar wavelets has high complexity. The fast Hankel transform algorithms in [18-19] for calculating electric field's Green's function in microstrip antennas is similar to Siegman, in that it requires exponential sampling and hence for fast changing functions can give rise to large errors. Hence from the above review, it is found that Higgins and Munson's method [5] is the best (in terms of computation time) and is chosen as the starting method for the work. According to [5], Hankel transform ($f_n(\rho)$) of a function($F_n(r)$) can be written as:

$$\phi_n(\eta) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty} |r| . \overline{F_n}(r) e^{jr\eta} dr \qquad (1)$$

$$f_n(\rho) = \begin{cases} 2\int_0^{\pi/2} \phi_n(\rho\sin\psi)\,\mathrm{Re}[e^{-jn\psi}]d\psi & n \ even \\ 2j\int_0^{\pi/2} \phi_n(\rho\sin\psi)\,\mathrm{Im}[e^{-jn\psi}]d\psi & n \ odd \end{cases} \quad (2)$$

where,

$$\overline{F_n}(r) = \begin{cases} F_n(r) & r \ge 0 \\ (-1)^n F_n(-r) & r < 0 \end{cases} \quad (3)$$

In digital applications, (1-3) are broken into discrete form. The number of angles between 0 to π/2 in (2) can be chosen to be less than the number of data points which reduces the computation burden somewhat. In discretized form, equations (1-3) can be written as:

$$\overline{\phi_n}(\eta_i) = \frac{\Delta r}{4\pi^2}\sum_{l=0}^{2P-1}\overline{F_n}(r_l)lin(l)W(l)e^{j(2\pi il/2P)} \quad (4)$$

P is the number of samples of $\Phi_n(\eta)$ and,

$$\overline{F_n}(r_l) = \begin{cases} F_n(l\Delta r) & l = 0,\ldots T-1 \\ 0 & l = T,\ldots.(2P-T) \\ (-1)^n F_n((2P-l)\Delta r) & l = (2P-T+1),\ldots..(2P-1) \end{cases} \quad (5)$$

T is the number of samples of $F_n(r)$

$$lin(l) = \begin{cases} r_l & l = 0,..P-1 \\ r_{2P-l} & l = P,..2P-1 \end{cases} \quad (6)$$

W(.) is a suitable window like Hamming, Hanning and so on.

$$f_n(\rho_m) = \begin{cases} 2\Delta\theta\sum_{k=0}^{S-1}\phi_n^I(k,m)\,\mathrm{Re}[ex(n,k)] & n \ even \\ j2\Delta\theta\sum_{k=0}^{S-1}\phi_n^I(k,m)\,\mathrm{Im}[ex(n,k)] & n \ odd \end{cases} \quad (7)$$

S is the number of samples of θ from 0 to π/2 with deviation Δθ and ex(n,k)=exp(-j[πn(k+.5)/2S])] and $\phi_n^I(k,m)$ corresponds to an interpolated version of $\phi_n(\rho_m \sin\theta_k)$. The modified input function (3) is multiplied by lin(l) (6) and then undergoes an FFT process in (4) to obtain samples of $\Phi_n(\eta)$ which are then integrated according to (7) to obtain samples of the Hankel transform $f_n(\rho)$.

III. METHODOLOGY, ARCHITECTURE AND ERROR ANALYSIS:

There are two parts to the architecture; the FFT section and the integration section. The detail of each section is explained as follows:

A. FFT BLOCK

The FFT section implements the Fourier transform via the radix-2 decimation in frequency (DIF) process [20]. Initially (5) and (6) are multiplied and the data is stored in memory. The data is then read out, complex exponential multiplication is performed by multipliers according to the FFT butterfly operation and the results are written back. The 'sin' and 'cos' values that represent the complex exponential of the twiddle factor are stored in look up table (LUT) from where they are read out for multiplication. The address references for a 256 point DIF radix-2 FFT (first two stages) are shown in Fig. 1 which is novel and directly mappable to hardware architecture for real time applications.

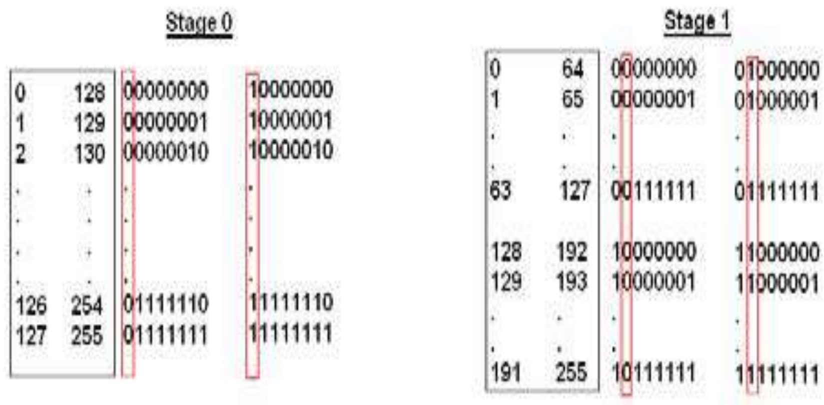

Fig. 1 Address references for first two stages of a 256 point radix 2 DIF FFT

In Fig. 1, the two adjacent numbers represent the addresses of the two elements involved in butterfly operation. From architectural point of view, they are the output of a 7-bit counter with a 0 or 1 (0 for the first element of the pair in butterfly operation and 1 for the second element) whose position depends on the stage number starting from the msb side. It can be implemented by a 7-bit counter and a set of multiplexers (MUXes). The angles are in arithmetic progression with a common difference of $2^{stagenumber}$. Hence the stage counter output after passing through a decoder is fed to the accumulator, whose output serves as the address for reading the 'sin' and 'cos' values of the required twiddle factor. The overall FFT structure is shown in Fig. 2.

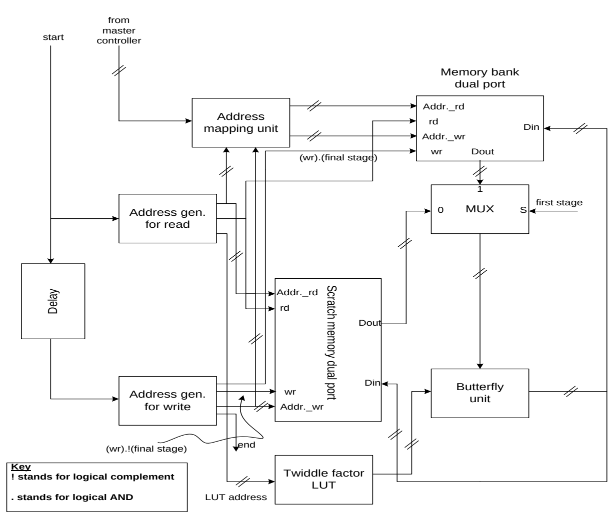

Fig.2 Architecture of the FFT block

In Fig. 2, initially, data is read from the memory, and the results of the intermediate stages are stored in scratch pad memory. Finally the results are written back to the memory. The address and angle generation blocks generate the input arguments to the butterfly unit that computes the twiddle factor multiplication. The later is derived from the address generation block itself as mentioned before.

B.THE INTEGRATION BLOCKS:

The method used is numerical quadrature. The integration block performs a set of sequential operations according to (7), which is basically the complex multiplications (performed by CORDIC) on the output of FFT operations (stored in memory), and then summations to obtain the result for different values of the transformed variable. In architecture, the sequence is carried out by a one-hot decoder that can be realized by a counter followed by a decoder. Initially, in memory, the set of values of $\Delta\rho\sin\theta_k$ for k =0 to S-1 is stored (say R). The angle is represented with positional weights from π, $\pi/2$, …, $\pi/2^n$. This technique helps in encoding the angles in binary sequence. In another register (say Q), the value of $n\Delta\theta$ is stored with n being the order of the Hankel transform. Another set of S memory locations (say R_1) is cleared. The steps are carried out in the following manner:

1. A counter (say i) is cleared to zero
2. Two registers (say S_1 and N) are cleared to zero
3. A counter (say j) is cleared to zero
4. Memory location j in R_1 is read out
5. To this value, 0.5 is added and the integer portion is used to reference the value of $\Phi()$ (i.e. the FFT output through nearest neighborhood interpolation)
6. The value is fed to a CORDIC block (the real part first, then the imaginary part) with the angle argument as N
7. At the same time, N is added with Q and the value is stored back in N. Also $R_1(j)$ is added with R(j) and the result is stored back in $R_1(j)$
8. Depending on n, the sine and cosine outputs are taken (by MUX) and the result is added with S_1
9. Increment j by 1 and repeat from iv. otherwise if terminal count of j is reached then write the content of S_1 to memory , increment i and repeat from ii. If i has reached terminal count then stop the process.

The CORDIC processor works in forward rotation mode. The angle can be any value from 0 to 2π, and hence depending on the quadrant, the angle is complemented (subtracted from $\pi/2$) as well as the final output is 2’s complemented. The 4-quadrant CORDIC follows a similar architecture mentioned in [20]. However, since each individual data value in the numerical quadrature of (7) is scaled by the same factor, the scale factor compensation is not required which simplifies the architecture compared to [20], and reduces the total number of iterations performed. Computation time can be improved further by selecting complex but fast adders like conditional sum adder [21] (to implement the add/sub operation inside CORDIC) that reduce the carry propagation time to logarithmic order. Schematic of a 4-bit conditional sum adder is shown in Fig. 3.

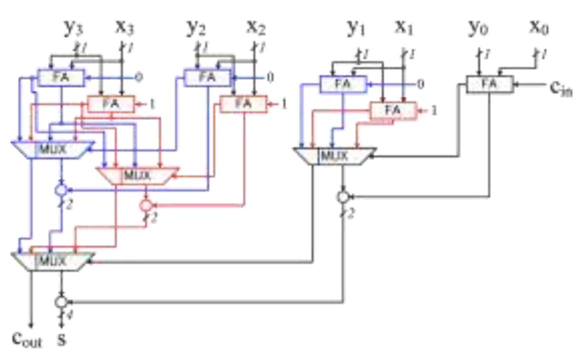

Fig. 3 Schematic of 4-bit conditional sum adder

International Journal of Education and Science Research Review
Volume-3, Issue-1 February- 2016 E-ISSN 2348-6457
www.ijesrr.org Email- editor@ijesrr.org
However, the CORDIC introduces its own error (which is now shown to be less than the method's inherent error). Since the scale factor computation is not present, the main error sources are the angle approximation errors and the truncation errors [22]. The mean square error (MSE) for angle approximation (E_{ang}) is given for cosine and sine cases respectively as :

$$E_{ang} = -\delta . v(0) . \sin(\beta) \tag{8a}$$

$$E_{ang} = \delta . v(0) . \cos(\beta) \tag{8b}$$

In (8), δ is the final residual angle, v(0) is the input vector and β is the rotated angle. The expression is a slight modification from [22]. The MSE for bit truncation is similarly given as (E_{trun}):

$$E_{trun} = 2\left(w_v(N^{'}-1) + \sum_{j=1}^{N^{'}-1} \prod_{i=j}^{N^{'}-1} w_v(j-1) k(i)^2 \right) + \left| E\{e_r(N^{'})\} + \sum_{j=1}^{N^{'}-1} \tilde{P}(j) E\{e_r(j)\} \right|^2 \tag{9}$$

In (9), $w_v(i)$ is the variance of truncation error at the ith stage from 0 to $N^{'}-1$ ($N^{'}$ is the number of stages in the CORDIC block) ; P(i) is the micro-rotation vector at ith stage given by $\begin{bmatrix} 1 & \sigma(i)2^{-i} \\ -\sigma(i)2^{-i} & 1 \end{bmatrix}$, where σ(i) is a sequence of ±1s; k(i)= $\sqrt{1+2^{-2i}}$; $e_r(i)$ is the truncation error at the ith stage and $\tilde{P}(i) = \prod_{j=i}^{N^{'}-1} P(j)$. The total error is the square root of the sum of squares of (8) and (9). However, for large bit lengths, the truncation error is much less than the angle error and is not considered . In the integration step of [5], there are S additions. Usually the CORDIC angle errors are uncorrelated and the power terms are added. However due to small value of S (about 128), the cross correlation terms may not cancel fully and the square of the sum of errors is taken (if the square is not taken, the sum denotes the mean error). This is finally multiplied by 2Δθ. Finally the square root of the mean of the squares of the errors of each transformed value gives the final root mean square error of the system relative to the multiplier based system. The error of the Hankel transform method [5] in approximating (2) by midpoint integration rule, for n even and n odd is respectively given as:

$$\frac{\pi\Delta\theta^2}{48} \left| \frac{\partial^2}{\partial\theta^2} \phi(\rho\sin\theta)\cos(n\theta) \right|_{\substack{\max \\ 0\le\theta\le\pi/2}} \tag{10a}$$

$$\frac{\pi\Delta\theta^2}{48} \left| \frac{\partial^2}{\partial\theta^2} \phi(\rho\sin\theta)\sin(n\theta) \right|_{\substack{\max \\ 0\le\theta\le\pi/2}} \tag{10b}$$

Differentiating (10), 2 times w.r.t. θ, gives the quadrature error for a given order to be proportional to ρ^2. Now, considering the proportionality constant to be 0.5 (heuristically) for both cases, the average error (e_{avg}) for both n even and odd can be approximately written as :

$$e_{avg} = \frac{\pi\Delta\theta^2}{48} . 0.5 . \frac{\Delta\rho^2}{\Delta\theta^2} . \frac{(L-1)^2}{3} \tag{11}$$

In (11), Δρ is the transformed variable sample deviation, and L is its number of samples.

IV. RESULTS

To test the CORDIC error relative to the error of the transform algorithm, the sinc function is employed of the form sin(ar)/ar, where a=2/(3Δr) with Δr=1 , which has a well defined Hankel transform given in [5]. The results for n=6 is shown in Fig. 4. Some values are shown in Table I. The CORDIC used is a 8-stage one with data width 40 bits (16 bits before decimal point, 24 bits after; 2's complement representation used). The number of input samples is 256 and the number of samples of the transformed variable is 128. The value of S in (7) is also 128. $\Delta\rho=2\Delta\theta=2\pi/256$; $\Delta r=1$;$\Delta\eta=0.0123$.

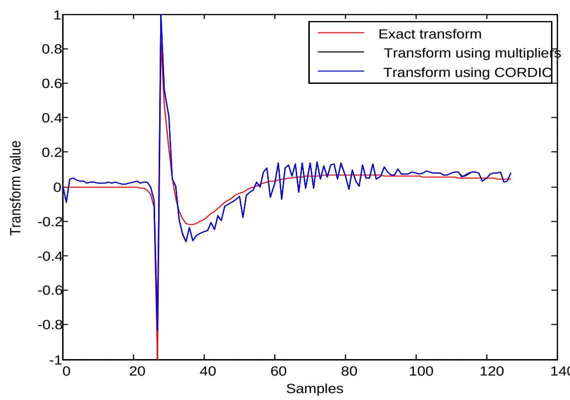

Fig. 4 Exact (analytical), computed using multipliers and computed using CORDIC (8 stages) 6th order Hankel transforms for sinc function

Table I : Some representative values of Hankel transform of sinc function (n=6) from Fig. 4

Sample No.	**Value (exact transform)**	**Value (by multipliers)**	**Value (by 8-stage CORDIC)**
0	0	0.0004	0.0004
4	0	0.0404	0.0405
18	-0.0008	0.0162	0.0163
24	-0.0214	0.0246	0.0246
38	-0.2150	-0.2829	-0.2822
46	-0.0932	-0.1133	-0.1127
53	-0.0109	-0.0324	-0.0325
67	0.0556	-0.0297	-0.0298
78	0.0662	0.0439	0.0445
95	0.0620	0.0994	0.0996

From Fig. 4, it is observed that the error with the use of CORDIC is minimal compared to the error of the algorithm. The square root of the mean square error between the ideal and computed using multipliers is 0.0554 while that between multipliers and CORDIC is 0.000455. The ratio is 121.76. The theoretical error between exact and computed using multipliers is 0.1076. The simulated result of the error between the exact transform and computed using multipliers is lesser than its theoretical value as a conservative condition had been considered for the latter, however, they are of the same order. The theoretical CORDIC error is 0.000458 which matches closely with simulation. It also shows that the CORDIC truncation errors are much less than the angle error. The results for the 9th order transform are shown in Fig. 5 and Table II.

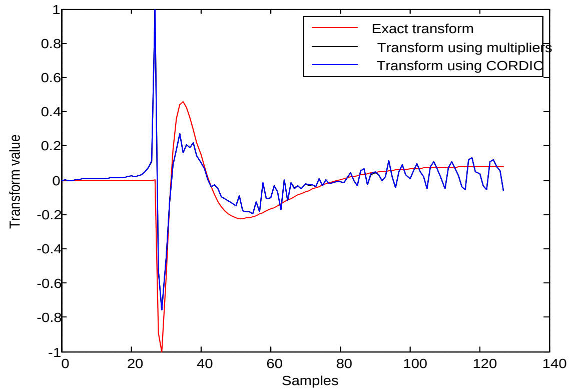

Fig. 5 Exact (analytical), computed using multipliers and computed using CORDIC (8 stages) 9th order Hankel transforms for sinc function

Table II : Some representative values of Hankel transform of sinc function (n=9) from Fig. 5

Sample No.	Value (exact transform)	Value (by multipliers)	Value (by 8-stage CORDIC)
0	0	0	0
4	0	0.0030	0.0030
18	0	0.0141	0.0142
24	0	0.0474	0.0475
38	0.2965	0.2174	0.2179
46	-0.1546	-0.0946	-0.0950
53	-0.2220	-0.1830	-0.1833
67	-0.0960	-0.0463	-0.0474
78	-0.0083	-0.0144	-0.0132
95	0.0568	0.0222	0.0225

The square root of the mean square error between the ideal and computed using multipliers is 0.1188 while that between multipliers and CORDIC is 0.000470. The ratio is 252.71, which again shows that CORDIC introduces minimal error compared to the error of the algorithm. The theoretical error between ideal transform and transform using multipliers is 0.1076, which is again of the same order as the value obtained from simulation. The theoretical CORDIC error is 0.000484, which matches closely with simulation.

V. CASE STUDY: APPLICATION IN A COMPUTER AIDED TOMOGRAPHIC SYSTEM:

The CORDIC based Hankel transform has been used in a complete tomographic image reconstruction system. Such systems are of use in medical fields to see the internal cross-section of an organ and also in industry to view the internal cross-section of a machine part. The method used is of [1] which is briefly outlined below :
The input to the method is 1-d projections of a cross section at a certain distance in the z-axis taken at different angles from 0 to 180° by parallel beam of x-rays. For our case, they have been generated synthetically using the radon transform (MATLAB's radon function used). Once the projections are obtained, they are Fourier transformed to obtain slices of the transformed image by the projection slice theorem. It is denoted as

$F(r,\theta)$ (r and θ are the radial and angular variable in the transform domain respectively). Now, $F(r,\theta)$ being periodic in θ with 2π can be broken into Fourier series in θ as :

$$F(r,\theta)=\sum_{n=-\infty}^{\infty}F_n(r)e^{-jn\theta} \qquad (12)$$

where the Fourier series coefficient $F_n(r)$ is given as ,

$$F_n(r)=\frac{1}{2\pi}\int_{-\pi}^{\pi}F(r,\theta)e^{jn\theta}d\theta \qquad (13)$$

The reconstructed image in polar domain $f(\rho,\Phi)$ also being periodic in Φ with period 2π can be expanded into Fourier series in Φ as :

$$f(\rho,\frac{\pi}{2}-\phi)=\sum_{n=-\infty}^{\infty}f_n(\rho)e^{-jn\phi} \qquad (14)$$

where, analogous to (13),

$$f_n(\rho)=\frac{1}{2\pi}\int_{-\pi}^{\pi}f(\rho,\frac{\pi}{2}-\phi)e^{jn\phi}d\phi \qquad (15)$$

Now, the inverse Fourier transform of each term in the Fourier series of (12) equals the corresponding term in the Fourier series of (14) and that by the definition of Hankel transform, $f_n(\rho)$ is the nth order Hankel transform of $F_n(r)$. It is expressed as :

$$f_n(\rho)=\frac{1}{2\pi}\int_{0}^{\infty}rF_n(r)J(r\rho)dr \qquad (16)$$

The Hankel transform is solved by (1-7), where CORDIC algorithm is used to solve (7), which is the most time consuming operation. The reconstructed image obtained is in polar domain. It is converted to Cartesian domain and the missing pixels' intensity values are calculated by bilinear interpolation [23]. The factor $\pi/2$ comes in (14) in order to get the proper form of Hankel transform equation (16). It results in the final image being rotated counter-clockwise by $\pi/2$.

The method has been used to recreate a 2-d cross section of human head. The head cross section model used is the Shepp and Logan head phantom [24]. The reconstruction using Bessel functions (16) , (1-7) using multipliers and (1-7) using CORDIC algorithm are shown in Fig. 6a, Fig. 6b and Fig. 6c respectively. The bit length is taken to be 40 bit (16 bits before the decimal and 24 bits after; 2's complement representation used) and the number of stages in CORDIC algorithm is taken as 12.The number of slices is taken as 128. The number of points in each slice is 256. The number of points of the transformed variable (ρ) for each Φ is 256, the number of terms in the Fourier series is 128 and number of summation terms in (7) is 64. $\Delta\rho=0.1047$; $\Delta\theta=\pi/128$; $\Delta r=0.1$; $\Delta\eta=0.2454$,

Fig. 6a Reconstruction using Bessel functions (modified Shepp Logan head phantom)

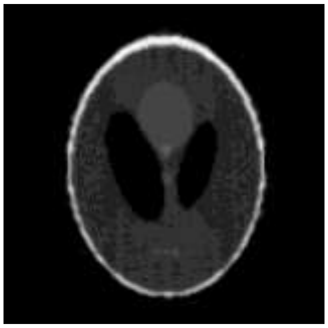

Fig. 6b Reconstruction using multipliers (modified Shepp Logan head phantom)

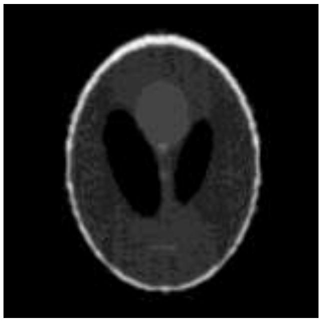

Fig. 6c Reconstruction using CORDIC (modified Shepp Logan head phantom)

The PSNR between Fig. 6a and Fig. 6b is 32.17 dB and between Fig. 6b and Fig. 6c is 56.66 dB (very close match for the latter). It shows that CORDIC introduces negligible error compared to [5]'s inherent error, while giving computation time improvement. If better quality is desired then it can be obtained by improving the contrast (e.g. histogram equalization) that takes negligible overhead.

For another example, square images (images with sharp edges) have been reconstructed. Images with discontinuities are present in various machine parts where tomographic image reconstruction may be used. The reconstruction using Bessel function (16) , (1-7) using multipliers and (1-7) using CORDIC are shown in Fig. 7a, Fig. 7b and Fig. 7c respectively. The parameter values are same as before.

Fig. 7a Reconstruction using Bessel functions (2 squares)

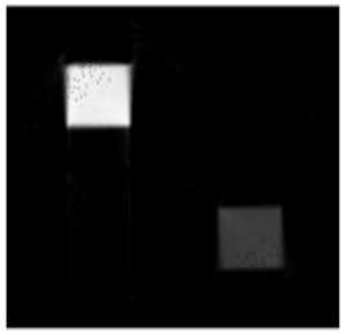

Fig. 7b Reconstruction using multipliers (2 squares)

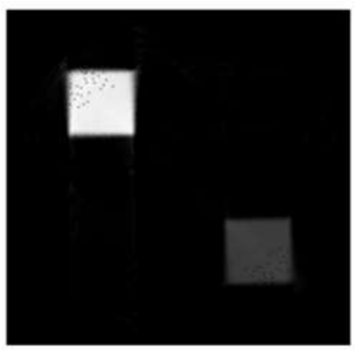

Fig. 7c Reconstruction using CORDIC (2 squares)

The PSNR between Fig.7a and Fig. 7b is 35.66 dB while that between Fig.7b and Fig.7c is 55.42 dB, once again illustrating the proposed methodology.

For 3-d reconstructions, multiple Hankel transforms of a given order are calculated for the 3rd dimension. The total fractional contribution of the CORDIC error remains same. The proposed method can be applied to the reconstructions in [25].

It is interesting to analyze the effect of noisy projections on the reconstructions. Noise in projections comes from lateral spreading of X-ray beam; small misalignment between the source and the detector. This has been modeled as zero mean additive white Gaussian noise. It has been found that for standard deviations upto 0.03, acceptable reconstructions are obtained with minimum error introduction by CORDIC algorithm as mentioned previously. The noise is smeared in concentric circles about the image (because of the smearing nature present in the backprojection operation). The results for the Shepp and Logan head phantom and squares for different noise standard deviations are shown in Fig. 8 and Fig. 9.

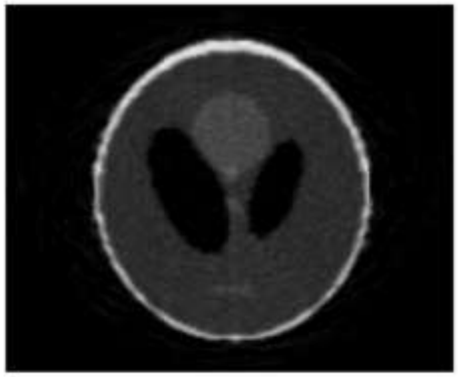

Fig. 8 a. Reconstruction of modified Shepp Logan head phantom in presence of zero mean additive white Gaussian noise having standard deviation 0.03.

The PSNR between Fig. 8 a and when there is no input noise is 35.35 dB.

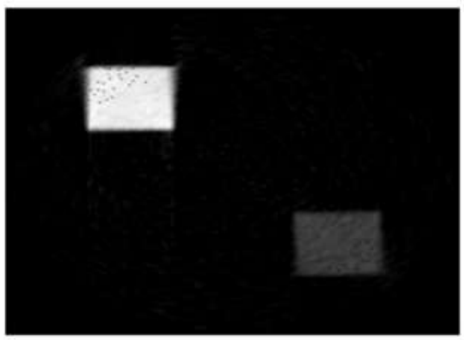

Fig. 8 b. Reconstruction of two squares in presence of zero mean additive white Gaussian noise having standard deviation 0.03.

The PSNR between Fig. 8 b and when there is no input noise is 33.55 dB.

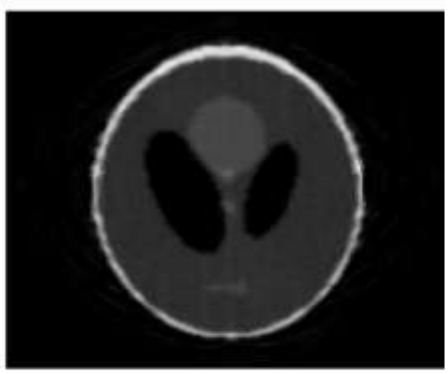

Fig. 9 a. Reconstruction of modified Shepp Logan head phantom in presence of zero mean additive white Gaussian noise having standard deviation 0.01.

The PSNR between Fig. 9 a and when there is no input noise is 45.09 dB.

International Journal of Education and Science Research Review
Volume-3, Issue-1 February- 2016 E-ISSN 2348-6457
www.ijesrr.org Email- editor@ijesrr.org

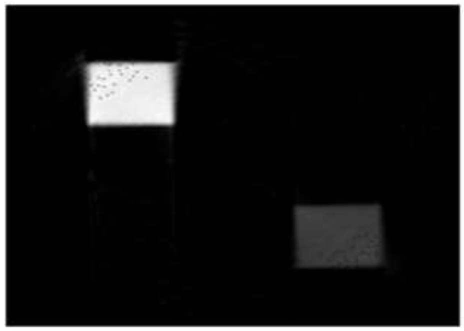

Fig. 9 b. Reconstruction of two squares in presence of zero mean additive white Gaussian noise having standard deviation 0.01.

The PSNR between Fig. 9 b and when there is no input noise is 42.31 dB.
It is seen that with noise standard deviations upto 0.03, PSNR values > 30 dB is obtained.
The computational cost has been compared with some state-of-the-art works in Table III.

Table III : Comparison with state-of-the-art

Work	Computational cost(real multiplications)
[5]	$2W.C.(4N'').\log_2(4N'')+2W.C.N''^2$
[9]	$2W.C.(4N'').\log_2(4N'')+2W.C.N''^2$
[13]	$2W.C.(4N'').\log_2(4N'')+2W.C.N''^2$
[17]	$4.W.C.N''^2$
This work	$2W.C.(4N'').\log_2(4N'')+2.12.C.N''^2$

In Table III, W is the word length (40-bit), C is the word-length-adder delay and N'' is the number of samples of the transformed variable (128). The CORDIC used is a 12 stage one (increasing the number of stages will only improve its error). From Table III, it is seen that by use of CORDIC, for the above parameters, 54.63 % savings in computation cost is obtained over the state-of-the-art with negligible error introduction. For the image reconstruction system, the time savings is 35.00 % (for the chosen parameter values given in Section-V) with same negligible error introduction. The computation time for FFT can be reduced by employing higher-radix FFTs.

VI. CONCLUSION:

Several applications require fast computation of Hankel transforms. A method and architecture for fast computation of Hankel transforms using the CORDIC algorithm is presented (based on a filtered back-projection method). The architecture is fast compared to that used by multipliers while the error contribution is minimal. Further since all the components are scaled by the same factor, the components need not be multiplied by the inverse CORDIC scale factor which again reduces computation time. Hence the proposed architecture is very much suitable for real time applications. Theoretical results have been substantiated by simulations. Real life application of the proposed methodology has been presented showing good correspondence to theory.

REFERENCES:

1. W.E. Higgins, And D.C. Munson Jr., "A Hankel Transform Approach To Tomographic Image Reconstruction", Ieee Trans. Med. Imag., Vol. 7, No. 1, Pp. 59-72, Mar. 1988.
2. R.J. Plemmons And V.P. Pauca, "Some Computational Problems Arising In Adaptive Optics Imaging Systems", Jnl. Of Computational And Applied Mathematics, Vol. 123, No. 1-2, Pp.467-487, Nov. 2000.
3. D. C. Munson Jr., J. D. O'brien, And W. K. Jenkins, "A Tomographic Formulation Of Spotlight-Mode Synthetic Aperture Radar", Proc. Ieee, Vol. 71, Pp. 917-925, Aug. 1983.
4. G. Yan, J.Tian, S. Zhu, Y. Dai, And C. Qin, "Fast Cone-Beam Ct Image Reconstruction Using Gpu Hardware", Jnl. Of X-Ray Sc. And Tech., Vol. 16, Pp.225-234, 2008.
5. W.E. Higgins, And D.C. Munson Jr., ''An Algorithm For Computing General Integer-Order Hankel Transforms'', Ieee Trans. Acoust., Speech, Signal Process., Vol. Assp-35,No. 1,Pp. 86-97, Jan. 1987.
6. J.E. Volder, "The Cordic Trigonometric Computing Technique", Ire Trans. Electronic Computers, Vol. Ec-8, No. 3, Pp.330-334,Sept. 1959
7. A.E. Siegman, "Quasi Fast Hankel Transform", Opt. Lett. Vol. 1, Pp. 13-15, 1977.
8. A.V. Oppenheim, G.V. Frisk And D.R. Martinez, "Computation Of The Hankel Transform Using Projections", J. Acoust. Soc. Am., Vol. 68, Pp. 523-529, Aug. 1980.
9. A.V. Oppenheim, G.V. Frisk And D.R. Martinez, "An Algorithm For The Numerical Evaluation Of The Hankel Transform", Proc. Ieee, Vol.66, No. 2, Pp. 264-265, Feb. 1978.
10. D. R. Mook, "An Algorithm For The Numerical Evaluation Of The Hankel And Abel Transforms", Ieee Trans. Acoust., Speech, Signal Process., Vol. Assp-31, No. 4, Pp. 979-985, Aug. 1983.
11. E.W. Hansen, "Fast Hankel Transform Algorithm", Ieee Trans. Acoust., Speech, Signal Process, Vol. Assp-33,No. 3, Pp. 666-671, Jun.1985.
12. S.M. Candel, "An Algorithm For The Fourier-Bessel Transform", Comp. Phys. Comm., Vol. 23, Pp. 343-353, 1981.
13. B.W. Suter, "Fast Nth-Order Hankel Transform Algorithm", Ieee Trans. Signal Process., Vol. 39, No. 2, Pp. 532-536, Feb. 1991.
14. V. Magni, G. Cerullo, And S. De Silvestri, "High-Accuracy Fast Hankel Transform For Optical Beam Propagation", J. Opt. Soc. Am. A., Vol. 9, No. 11, Pp. 2031-2033, Nov. 1992.
15. J.A. Ferrari, "Fast Hankel Transform Of Order 0", J. Opt. Soc. Am. A., Vol. 12, No. 8, Pp. 1812-1813, Aug. 1995.
16. L. Knockaert, "Fast Hankel Transform By Fast Sine And Cosine Transforms : The Mellin Connection", Ieee Trans. Signal Process., Vol. 48, No. 6, Pp. 1695-1701, Jun. 2000.
17. R.K. Pandey, V.K. Singh, And O.P.Singh, "An Improved Method For Computing Hankel Transform", Jnl. Of The Franklin Institute, Vol. 236, Pp. 102-111, 2009.
18. P.P. Ding, S. Zhoudi, L.W. Li, S.P.Yeo, N.B. Christensen, "A Modified Fast Hankel Transform Algorithm For Calculating Planar Multilayered Green's Function", In Proc. Electromagnetics In Advanced Appl. Iceaa 2010, Pp. 20-24, Sept. 2010.
19. S.Q. Li, C.H. Chan, L. Tsang, And C.C. Huang, "Closed-Form Spatial Electric Field Green'S Functions Of Microstrip Structures Using The Fast Hankel Transform And The Matrix Pencil Method", Iee Proc. Microw.,Antennas, Prop.,Vol. 147, No. 3, Pp. 161-166, Jun. 2000.
20. K.Hwang, "Computer Arithmetic : Principles, Architecture, And Design", New York, John Wiley & Sons, Inc., 1979, Pp. 78-81
21. S.Y.Park, And N.I.Cho, "Fixed Point Error Analysis Of Cordic Processor Based On Variance Propagation Formula", Ieee Trans. Circuits Syst. –I, Vol. 55, No.3, Pp. 573-584, Mar. 2004.
22. Bilinear Interpolation (2016 Jan.) Available : Http://En.Wikipedia.Org/Wiki/Bilinear_Interpolation
23. L. A. Shepp And B. F. Logan, "The Fourier Reconstruction Of A Head Section", Ieee Trans. Nucl. Sci., Vol. Ns-21, No. 3, Pp. 21-43, Jun. 1974.
24. Q. Tang, G.L. Zeng, J. Wu, And G.T. Gullberg, "Exact Fan-Beam And Cone-Beam Algorithms With Uniform Attenuation Correction", In Proc. 8th Int. Meeting On Fully Three Dimensional Image Reconstruction In Radiology And Nuclear Science,3d 05, Pp. 255-258, Jul. 2005.