A Single CORDIC Based Architecture for Real-time Tomographic Image Reconstruction for Industrial Applications

Amitava Ghosh
Department of Electronics & Communication Engineering
Institute of Engineering & Management
Kolkata, India
amitava.ghosh@iem.edu.in

Anindya .S. Dhar
Electronics & Electrical Communication Engineering Department
IIT Kharagpur
Kharagpur, India
asd@ece.iitkgp.ac.in

Indrajit Chakrabarti
Electronics & Electrical Communication Engineering Department
IIT Kharagpur
Kharagpur, India
indrajit@ece.iitkgp.ac.in

2024 IEEE Calcutta Conference (CALCON) | 979-8-3315-1666-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CALCON63337.2024.10914154

Abstract— **In this paper, a single co-ordinate rotation digital computer (CORDIC) based architecture for real-time tomographic image reconstruction is presented. The method is based on solving the Hankel transform of a function. Four out of the five steps of the algorithm are fast Fourier transform (FFT) reducible and a CORDIC based architecture can be followed. The Hankel transform is also broken into sine and cosine terms via the Chebyshev polynomial representation of the Bessel function. The method has O(N^3) complexity. A single CORDIC hardware is very easily realizable which makes the system extremely cheap. Signal to noise ratio (SNR) analysis of reconstructed images shows very little degradation (SNR degradation is less than 15% for reconstructed images). The computation time is also very fast compared to existing literature (0.022 secs on Virtex-6 FPGA; 0.095 secs on Spartan-3E FPGA and 2.56 secs on software platform), making it an extremely viable option.**

Keywords— ***Architecture, Computed tomography, CORDIC block, Fast Fourier transform, Hankel transform, Real-time signal processing***

I. Introduction

Computed tomography (CT) is required in industrial applications to view different machine parts and find out the flaws. In contrast to biomedical CT, here a large number of images need to be generated in a short span of time and reconstructed image quality need not be that high (as compared to biomedical domain) calling for real-time signal processing requirements. In this paper, we have achieved that using a single co-ordinate rotation digital computer (CORDIC) based architecture.

References [1-3] have developed reconstruction systems which are big and costly. We are designing a single CORDIC solution that would occupy low die area and hence be less costly and at the same time be simple. Reference [4] have considered iterative reconstruction techniques for image generation. However, their computation time is very slow about 11.89 secs for 30 iterations and it increases linearly with iteration number. Although, [4] have claimed to have removed streak artifacts but our signal to noise ratio (SNR) analysis show little degradation. Reference [5] have evaluated reconstruction times of 2-d generated CT images using CPU and graphics processing unit (GPU). GPU time was 0.52 secs; CPU with threads took 1.37 secs and sequential CPU took 4.69 secs all of which are significantly slower than our work. A more detailed comparative discussion of the various methods in CT is provided in Table I. So prior art fails to deliver a system that occupies low area, is cheap, shows very little image quality degradation, and is fast, which is the subject of the present paper.

The rest of the paper is organized as follows. In Section II the mathematical formulation behind the algorithm is presented. The architecture is described in Section III with diagrams. Results are given in Section IV with extensions to biomedical domain in Section V, while Section VI provides the conclusion.

II. Mathematical Formulation Behind The Algorithm

The mathematical formulation follows from [7]. It starts with a set of 1-d projections $p_\theta(t)$ which in our case have been obtained by the Radon transform on the original image. Fourier transform of $p_\theta(t)$ gives slices of the Fourier transform of the final image $F(r,\theta)$. Only half of them need to be retained because of the conjugate symmetry property of $F(r,\theta)$ as the image is real :

$$F(r,\theta) = F^*(r,\theta+\pi) \tag{1}$$

Now, $F(r,\theta)$ being periodic in θ with period 2π, can be expanded into a Fourier series in θ as :

$$F(r,\theta) = \sum_{n=-\infty}^{\infty} F_n(r).\, e^{-jn} \tag{2}$$

$F_n(r)$ is the n$^{\text{th}}$ order Fourier series coefficient which is expressed as :

$$F_n(r) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(r,\theta).\, e^{jn\theta} d\theta \tag{3}$$

$F_n(r)$ is obtained by a fast Fourier transform (FFT) in θ domain. The final image $f(\rho,\phi)$ being periodic in ϕ with period 2π, can also be expanded into a Fourier series :

$$f\left(\rho,\frac{\pi}{2}-\phi\right) = \sum_{n=-\infty}^{\infty} f_n(\rho).\, e^{jn\phi} \tag{4}$$

For negative values of $'n'$, we use the following relation :

$$f_n(\rho) = f_{-n}^*(\rho) \tag{5}$$

$f\left(\rho,\frac{\pi}{2}-\phi\right)$ is obtained by another FFT in $'n'$ domain. Analogously,

$$f_n(\rho) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f\left(\rho,\frac{\pi}{2}-\phi\right).\, e^{-jn\phi} d\phi \tag{6}$$

TABLE I. COMPARATIVE DISCUSSION OF THE VARIOUS METHODS EMPLOYED IN CT

Method	Description	Reconstruction time	Cost	Limitations	Application scenarios
Deep learning based [1]	Uses artificial intelligence to reconstruct very high quality images.	Moderate to high	High	Quality of the training data set must be very high.	Biomedical to high precision industrial CT.
Iterative reconstruction based [4], [6]	It begins with an image assumption and compares it to measured values while making constant adjustments until the two are in agreement. Image quality very high.	High	Moderate to high	Computation time increases linearly with iteration number (image quality).	Predominantly biomedical but also high precision industrial CT.
Convolution-backprojection method (CBP) [7]	Has a filtering stage followed by a backprojection operation. Image quality moderate to high.	Very low	Low	Affected by noise and streak artificats in lower dose CT.	Real time industrial CT.
Proposed method	Finds the polar form of the reconstructed image via a Hankel transform. Image quality moderate to high.	Lower than CBP method [7]	Low	Affected by noise and streak artificats in lower dose CT.	Real time industrial CT.

Now, the inverse two-dimensional Fourier transform of each term $F_n(r).e^{-jn\theta}$ is $f_n(\rho).e^{jn\phi}$. The relationship is nothing but a Hankel transform pair given by :

$$f_n(\rho) = \frac{1}{2\pi}\int_0^\infty r.F_n(r).J_n(r\rho)dr \quad (7)$$

In (7), $J_n(.)$ is the Bessel function of the first kind of order $'n'$. The factor $\pi/2$ in $f\left(\rho, \frac{\pi}{2} - \phi\right)$ is required to get the proper form of the Hankel transform equation. It results in $\pi/2$ radians of rotation of the final image. The Hankel transform equation in (7) is solved by the algorithm in [8] which formulates it as :

$$\widehat{F_n}(r) = \begin{cases} F_n(r) & r \geq 0 \\ (-1)^n F_n(-r) & r < 0 \end{cases} \quad (8)$$

$$\phi_n(\eta) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty} |r|.\widehat{F_n}(r).e^{jr\eta}dr \quad (9)$$

Equation (9) is analogous to a scaled inverse fast Fourier transform (IFFT) which is solved by an FFT block where the input and output values are conjugated. Finally,

$$f_n(\rho) = \begin{cases} 2\int_0^{\pi/2} \phi_n(\rho\sin\psi).\cos(n\psi)\,d\psi & n\ even \\ -2j\int_0^{\pi/2} \phi_n(\rho\sin\psi).\sin(n\psi)\,d\psi & n\ odd \end{cases} \quad (10)$$

The algorithm gives the final image in polar form which is directly displayable if a polar display is present [9]. Otherwise, polar to Cartesian conversion followed by interpolation is required to display the image. If all the parameters of the algorithm are same, then it has O(N^3) complexity. The total time for FFT computation is $4 \times N^2 \times log_2(2N)$ considering half of the operations require twiddle factor multiplication. The time for $f_n(\rho)$ computation is N^3. So, if N is increased by a factor of 2, the computation cost is increased by approximately 8 times. This increased burden might not be acceptable for large N. On the other hand, for small N the reconstructed image quality suffers. We give the average computation time for different N (on an Intel core i3 x64 processor running at 3.60 GHz and having 8 GB installed RAM) and also show what fraction of it is used in the $f_n(\rho)$ computation, in Table II.

III. ARCHITECTURE DESIGN

A. CORDIC block

The CORDIC block is at the heart of the system. A single CORDIC block performs the whole data processing operation. The CORDIC employed is a 4-quadrant pipelined one (in circular rotation mode) with a pre-rotation stage at the beginning to accommodate the remaining two quadrants. 4-quadrant CORDIC is required because in the $f_n(\rho)$ computation block for higher orders of 'n', the angular argument would be larger than 360°. Pipelined CORDIC is used as it increases the throughput of the system. Also, a non-pipelined version would require 2 barrel shifters that would occupy a large number of configurable logic blocks (CLBs). Circular rotation mode is used as in this mode sine and cosine values are obtained at the outputs. The CORDIC block follows the architecture presented in [10] and is depicted in Fig. 1. After the rotation operation, scaling operation is performed (new to our architecture) by a series of shift right operators and pipelined adders. In the $f_n(\rho)$ computation block, all the elements are uniformly multiplied by the scale factor so the latter operation is not done, it is bypassed by a MUX with select line as the flag bit in Fig. 1.

B. FFT block

Decimation in frequency (DIF) FFT algorithm has been chosen as the FFT algorithm. The choice of the CORDIC algorithm for implementing the elementary butterfly operation saves a lot of hardware without sacrificing speed. In fact, in our architecture, no sine/cosine values need to be

TABLE II. COMPUTATION TIMES FOR VARYING 'N'

N	Computation time (s)	Percentage of time used in the $f_n(\rho)$ computation (%)
512	302.92	92.37
256	16.83	86.89
128	3.85	79.78
64	0.29	68.82
32	0.086	56.16

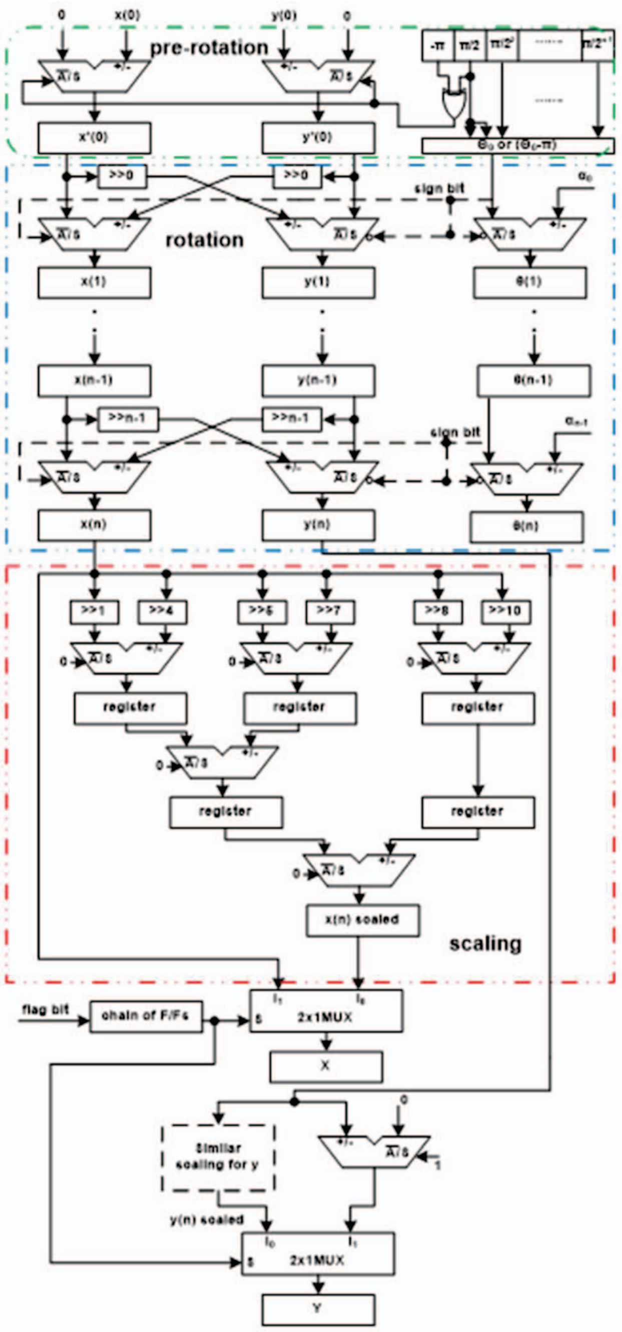

Fig.1. 4-quadrant pipelined CORDIC in circular rotation mode.

stored except for a very few number of arctangents. In FFT, the most error-affected operation is the CORDIC rotation. The CORDIC operation affects both the output elements of the basic butterfly unit in the decimation in time (DIT) FFT algorithm. While it affects only one output element of the basic butterfly unit in the DIF FFT algorithm. Hence the latter is more accurate and has been used in our work. The architecture of the FFT block followed is that of [11]. The only modification is : initial data is fetched from the memory bank and final stage data is written to the memory bank while intermediate stage data is stored in a scratch-pad memory. The FFT operation is required in four out of the five steps of the algorithm.

C. $f_n(\rho)$ computation block

Here at first $\rho_m sin\theta_k$ computation is carried out for referencing $\phi_n(\cdot)$ (the architecture is not shown in figure). This is performed by the CORDIC block with flag=1 (scale factor computation bypassed). Only $\Delta\rho sin\theta_k$ is computed by the CORDIC $\forall k$ and the integer multiples of it are obtained by accumulate and add operation. Since $\Delta\rho$ is constant so $\Delta\rho \times k_{scaleCORDIC}$ is pre-computed by the user before the start of the algorithm and stored in an input register, which is used as the X input to the CORDIC (Y input being 0). It not only reduces the latency of the CORDIC but also makes the operation less erroneous (a high precision value of $k_{scaleCORDIC}$ can be multiplied with $\Delta\rho$). This is of utmost importance as $\rho_m sin\theta_k$ is used for address referencing. The computation of $f_n(\rho_m)$ from $\phi_n(\cdot)$ is depicted in Fig. 2. Three counters correspond to the three loops and a mod-2 counter is required for treating the real and imaginary parts of $\phi_n(\cdot)$ separately. $\rho_m sin\theta_k$ is read out from memory, is used to reference $\phi_n(\cdot)$ and real and imaginary parts of it are fed separately to the CORDIC. The $sin\theta_k$ and $cos\theta_k$ terms are selectively taken according to the order of the Hankel transform and are accumulated in a register and finally written back to the memory by a write control circuitry. The latter consists of another three nested counters that are activated $'L'$ time units later where $'L'$ is the latency of the CORDIC block and the memory referencing for read delay. The CORDIC scale factor is not multiplied as all the terms are changed by it. It is multiplied later (more accurate version) through a fast multiplier whose complexity is O(N^2) and is less than that of the main algorithm.

The overall architecture directly follows from the discussion presented above.

IV. Results

The architecture has been modelled at bit level in MATLAB which includes bit shift and bit truncation effects. Data width is 32 bits for each real and imaginary part which is in line with modern signal processing systems [5]. Fixed point 2's complement arithmetic has been followed with 12 bits before the point and 20 bits after it as this eliminates chances of overflow and underflow (seen empirically). The parameters of the algorithm are : CORDIC length=14; P=T=L=128; S=64; Hankel transforms are computed from 0^{th} to 127^{th} order and the number of projections is 128 as most modern CT scanners are of such type. The choice of the above parameters gives good image quality (little SNR degradation) with little computational delay (tested empirically). Also, with these specs, the FFT size, in all the FFT-based steps, gets fixed to 256 with 128 such FFTs being performed. This makes the system more hardwired increasing speed and reducing area.

As examples to reconstruct industrial images, we consider three cases. The first one is a picture of 2 metal plates with different densities. The original image, reconstructed image with infinite precision, and reconstructed image with finite precision is shown in Fig. 3.

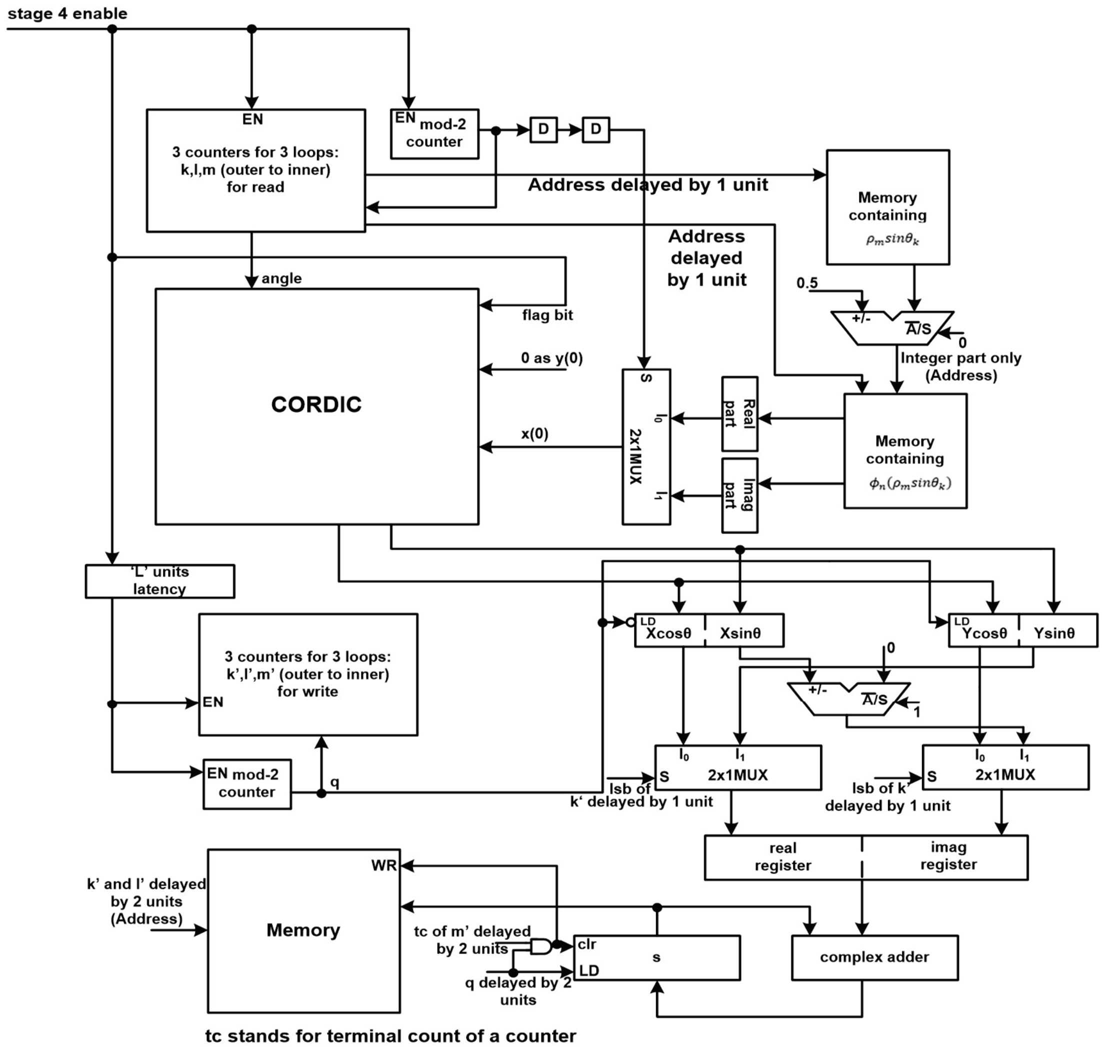

Fig.2. $f_n(\rho)$ computation block architecture.

Next is a square metal plate whose density varies uniformly from one edge to the opposite. The images are shown in Fig. 4.

The final image is a square metal plate with a crack. The same set consisting of the three images is depicted in Fig. 5.

From Fig. 3, 4 and 5 it is seen that the algorithm as well as the architecture reproduces the images with minimum degradation in SNR and their results are almost identical. Hence, we can say that the algorithm and architecture are working correctly.

The time taken by the architecture for computation (T_{comp}) is given by :

$$T_{comp} = T_{FFT} + T_{f_n(\rho)} + T_{oth} \tag{11}$$

T_{FFT} is the time taken for the FFT computation process and is given by :

$$T_{FFT} = (8 \times 256 + 21) \times 128 \times 4 \times T \tag{12}$$

256-point FFT is performed, so there are 8 stages. 21 cycles is the latency from the read to the write control circuitry activation. FFT is done 128 times per step and there are 4 steps. T is the clock period. $T_{f_n(\rho)}$ is the time taken for $f_n(\rho)$ computation and is given by :

$$T_{f_n(\rho)} = (128 \times 128 \times 64 \times 2 + 19) \times T \tag{13}$$

Hankel transforms are computed from 0th to 127th order; L=128; S=64; we have to rotate the real and imaginary parts separately so the factor of 2 is coming and 19 is the latency from the read to the write control circuitry activation in Fig. 2. T_{other} is the time taken for the low complexity operations like $\rho_m sin\theta_k$ obtainment, multiplication of CORDIC scale factor to $f_n(\rho)$ and generation of $|r|$ in the computation of $\phi_n(\eta)$ in (9). Neglecting it, the total computation time comes out as :

$$T_{comp} = 3164708 \times T \tag{14}$$

The critical path delay consists of : i. a 32-bit adder/subtractor delay, ii. a 2×1 MUX delay, iii Set-up and hold time of the flip-flops. The critical path delay has been simulated in Xilinx ISE for i. Virtex-6 FPGA; Device: XC6VLX75T; Package: FF784; Speed: -3 and it comes out to be 7ns; and a lower-end FPGA (for lower cost) ii. Spartan-3E FPGA; Device: XC3S1600E; Package: FG484; Speed: -5 and the critical path delay comes out to be 30.06ns. The total computation time is shown in Table III.

(a) (b) (c)

Fig.3. Two metal plates reconstruction case (a) Original image (b) Reconstructed image with infinite precision (algorithm) (c) Reconstructed image with finite precision (architecture).

(a) (b) (c)

Fig.4. Metal plate with uniform density variation reconstruction case (a) Original image (b) Reconstructed image with infinite precision (algorithm) (c) Reconstructed image with finite precision (architecture).

(a) (b) (c)

Fig.5. Metal plate with crack reconstruction case (a) Original image (b) Reconstructed image with infinite precision (algorithm) (c) Reconstructed image with finite precision (architecture).

TABLE III. COMPUTATION TIMES OF THE ARCHITECTURE ON DIFFERENT PLATFORMS

Computation time on Virtex-6 FPGA (s)	Computation time on Spartan-3E FPGA (s)	Computation time in MATLAB (s)
0.022	0.095	2.56

From Table III, it is seen that our architecture is very fast, in fact, it is the fastest amongst the current literature to the best of our knowledge.

Colored images can be obtained by operating the algorithm on individual RGB values. 3-d images can be reconstructed by considering 2-d transforms instead of 1-d.

The error due to CORDIC has been analyzed in [12]. This error is multiplied by twiddle factors according to the FFT butterfly and comes to the output along with the transformed samples. For the next stage of processing, we have to consider the previous stage error as input to it. For the $f_n(\rho)$ computation block, CORDIC errors are generated with each sine/cosine multiplication according to [12] and are accumulated. All the stage I-IV errors are input to stage V FFT which also generates its own inherent error and the total error contaminates the image. The absolute errors for the 3 reconstructed images generated by the proposed architecture are plotted in Fig. 6. The mean absolute error is very small in magnitude relative to the average absolute signal value which justifies the minimal degradation scenario.

V. EXTENSION TO BIOMEDICAL DOMAIN

The limitations of the current approach are : i. noise injection in images especially in lower-dose CT scans (which are used in biomedical field) and ii. streak artifacts. Since in biomedical field, the image quality must be high, these effects must be removed. The proposed method can be used with certain modifications which trade-off computation time with reconstructed image quality. Some of these new techniques to be explored are : i. proposed technique along with model-based iterative reconstruction methods ii. proposed technique along with deep learning-based methods iii. use of interpolation along with the reconstruction algorithm and iv. use of filtering in conjunction with the reconstruction algorithm to remove the sources of error.

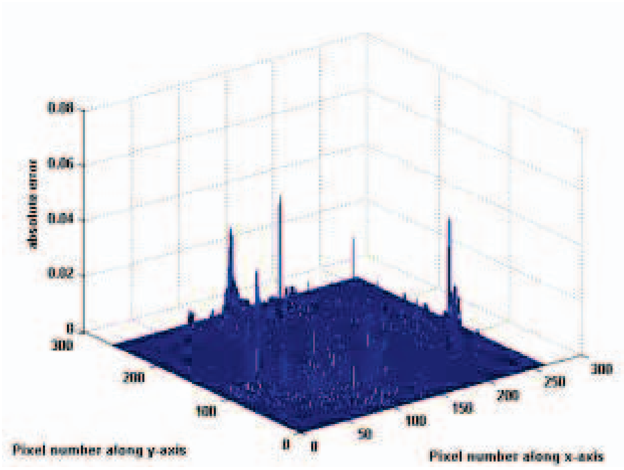

Fig. 6(a). Absolute error plot for the two metal plate reconstruction case. Average absolute signal value : 0.047; mean absolute error : 0.00047.

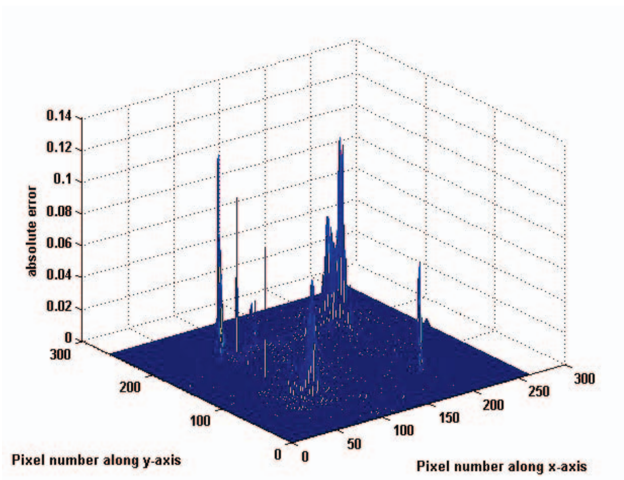

Fig. 6(b). Absolute error plot for metal plate with uniform density variation reconstruction case. Average absolute signal value : 0.128; mean absolute error : 0.00061.

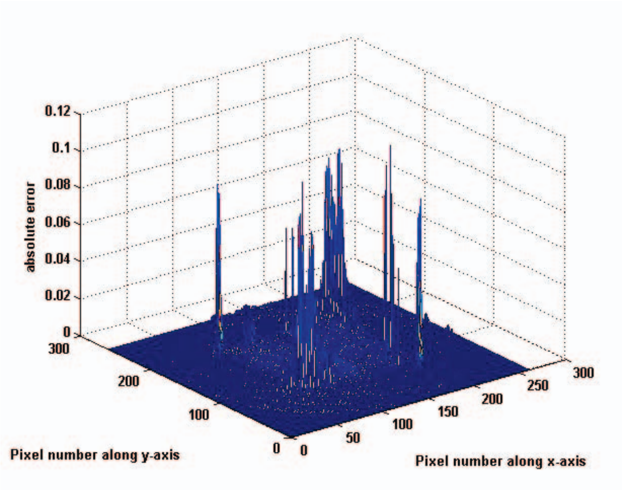

Fig. 6(c). Absolute error plot for metal plate with crack reconstruction case. Average absolute signal value : 0.161; mean absolute error : 0.0018.

VI. Conclusion

In this paper, an architecture of a Hankel transform based approach for computing CT images has been presented. The Hankel transform algorithm employed is projection based. The architecture uses a single CORDIC block in circular rotation mode for data processing operation. Industrial CT reconstructed images show that the SNR degradation is minimal. The critical path delay consists of only a 32-bit adder/subtractor delay; a 2×1 MUX delay and set-up and hold time of flip-flops resulting in very low reconstruction times. It can be easily used in the real-time reconstruction of images. Future work is to extend the application area to the biomedical domain.

References

[1] C. Zhang, G. Zhu and J. Fu, "A Deep Learning Reconstruction Method for Fast Assembly Line Computed Tomography," *2021 IEEE International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*, Qingdao, China, 2021, pp. 748-752, doi: 10.1109/ICEEMT52412.2021.9602188.

[2] Z. Zuo and T. Sun, "Application of Industrial CT Technology in Additive Manufacturing Field," *2022 Photonics & Electromagnetics Research Symposium (PIERS)*, Hangzhou, China, 2022, pp. 197-203, doi: 10.1109/PIERS55526.2022.9792748.

[3] Q. Fumin, "Design of an Electric Vehicle Modeling and Visualization System Based on Industrial CT and Mixed Reality Technology," *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, Dalian, China, 2023, pp. 429-433, doi: 10.1109/ICDSCA59871.2023.10393763.

[4] R. Acharya, U. Kumar, V. H. Patankar, S. Kar and A. Dash, "Reducing Metal Artifact using Iterative Reconstruction in Industrial CT," *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, Navi Mumbai, India, 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487687.

[5] A. Cordeiro, P. Filho, H. Silva, A. Candido Junior, E. Casanova and J. Spancerski, "Performance evaluation in the reconstruction of 2D images of computed tomography using massively parallel programming CUDA" in *Brazilian Journal of Development*, vol. 8, no. 2, pp. 10104-10116, 2022, https://doi.org/10.34117/bjdv8n2-110

[6] Iterative Reconstruction (CT), [Online]. Available: https://radiopaedia.org/articles/iterative-reconstruction-ct , Accessed Oct. 2024

[7] W. E. Higgins and D. C. Munson, "A Hankel transform approach to tomographic image reconstruction," in *IEEE Transactions on Medical Imaging*, vol. 7, no. 1, pp. 59-72, March 1988, doi: 10.1109/42.3929.

[8] W. Higgins and D. Munson, "An algorithm for computing general integer-order Hankel transforms," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 1, pp. 86-97, January 1987, doi: 10.1109/TASSP.1987.1165032.

[9] C. F. R. Weiman and G. Chaikin, "Logarithmic spiral grids for image processing and display," *Comput. Graphics Image Processing*, vol. 11, issue 3, pp. 197-226, November 1979, https://doi.org/10.1016/0146-664X(79)90089-3

[10] T. Kulshreshtha and A.S. Dhar, "CORDIC-based Hann windowed sliding DFT architecture for real-time spectrum analysis with bounded error-accumulation," in *IET Circuits Devices Syst.*, vol. 11, pp. 487-495, July 2017, https://doi.org/10.1049/iet-cds.2016.0375

[11] A. Banerjee, A.S. Dhar and S. Banerjee, "FPGA realization of a CORDIC based FFT processor for biomedical signal processing," in *Microprocessors and Microsystems*, vol. 25, issue 3, pp. 131-142, May 2001, https://doi.org/10.1016/S0141-9331(01)00106-5.

[12] S.Y. Park and N.I. Cho, "Fixed point error analysis of CORDIC processor based on the variance propagation," *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, Baltimore, MD, USA, 2003, pp. II-833, doi: 10.1109/ICME.2003.1221746.