A REGULAR ALGORITHM FOR REAL TIME RADON & INVERSE RADON TRANSFORM

Abhishek Mitra and Swapna Banerjee

E & ECE dept, IIT Kharagpur 721302, India e-mail: mitra_avi@indiatimes.com, swapna@ece.iitkgp.ernet.in

ABSTRACT

To meet the real time requirements an interpolation free, parallel algorithm for the Fast Radon Transform (FRT) and Inverse FRT (IFRT) is proposed. The proposed method solves all the important problems associated with the previous interpolation free FRT and IFRT algorithm and reduces the number of computations and algorithmic complexities significantly. The proposed algorithm is highly regular and we also describe a methodology to design a dedicated parallel processing architecture from the view point of its efficient implementation.

I. INTRODUCTION

The Radon Transform of an image is a set of projection of the image taken at different angles [1]. By means of the Radon transform one can determine a system internal structure without physically probing the interior. For this reason Radon transform has been adopted in widespread applications such as tomography, ultrasound, x-ray, nuclear magnetic resonance imaging, optics, stress analysis, geophysics and many others. In these applications, DRT of the object can be easily obtained by the projections of x-ray, ultrasound or similar projecting sources [1] [2]. The Inverse DRT to retrieve the original image, faces a lot of problems in realization. One popular inversion method is the 'Filtered back projection algorithm' [1]. The various difficulties associated with this algorithm are, first, the conversion between radial coordinates and a raster scan format, second, the interpolation required to compute the line integral approximations on a rectangular grid, and finally, the significant computational requirements are necessary to calculate the inverse, however its VLSI implementation was also tried [3]. To solve these difficulties Beylkin's proposed a distinctly different approach for the DRT in τ-p domain [4]. Based on this approach, Kelley and Madisetti further proposed an improved Fast Radon Transform (FRT) and the Inverse Fast Radon Transform (IFRT) algorithm [5]. However, this algorithm suffers from large number of computations because of zero padding, and also very complex because of its three different flows of computations which is not suitable for implementation in hardware [6]. Another interpolation free algorithm was proposed by Lun in 1995 [7]. Although that was a multiplication free algorithm but it does not facilitate parallel processing and requires a large number of additions and also consists of a dense projections near 0^0 and 90^0 whereas a less number of projections near 45^0. In this paper, a new method for the FRT and IFRT is proposed based on the Kelley-Madisetti algorithm. The proposed method successfully overcomes these problems resulting in a much reduction in number of computation followed by a simple algorithm and therefore acts as a basis for a compact implementation in hardware to meet the real time requirements. Section II introduces the proposed algorithm and a methodology to design a fast hardware architecture of the proposed algorithm. In section III an example has been presented and an analysis to prove the efficiency of the proposed method. Section IV concludes the paper.

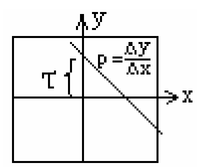
II. PROPOSED METHOD

Let u(t,x) be a 2-d image (ref. to fig.1), then its continuous Radon-Transform (R-T) is given by [1]:

$$R\{u(t,x)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(t,x)\,\delta(t-px-\tau)\,dx\,dt \qquad \text{……..(i)}$$

R-T of an image is the integral of the image intensities along different possible straight lines defined by the parameters τ and p through the image (fig.1). Corresponding Radon space is constructed with two parameters τ and p.

The number of computations reduces significantly when this space domain computation is computed in the frequency domain. Let $t'=t-\tau$, one can rewrite eqn.(i) follows:



$$R\{u(t,x)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(t',x)\,\delta(t'-px)\,dx\,dt'$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(t',x)\,\delta(t',px)\,dx\,dt' \qquad \text{…..(ii)}$$

Fig.1 An Image with a st. line

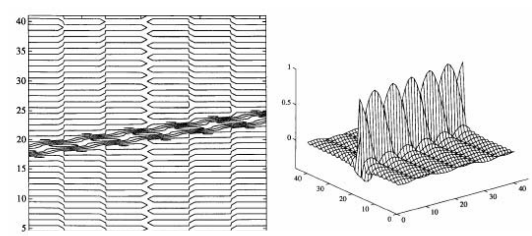


Fig.2 Line δ function for the angle 8.05^o

In equation (ii) unit impulse function is replaced by two-dimensional line impulse function (fig.2):

$$\delta(n,m) = \begin{cases} \propto & \text{if } (n-m)^2=0 \\ 0 & \text{if } (n-m)^2\neq 0 \end{cases} \text{ and}$$

$$\int_{-\alpha}^{\alpha}\partial(n-m)dn = 1, \quad \int_{-\alpha}^{\alpha}\partial(n-m)dm = 1 \quad \text{for } n, m \in \Re \quad \text{(iii)}$$

$\Re$ is the set of real numbers. Impulse function in (ii) performs only the rotation operation while the input image is translated.

Discrete Radon Transform:

Assume that u(t,x) is of finite support. Let y(n,s) represents the discrete approximation to R{u(t,x)}, and let x(m,*l*) represents the discrete version of u(t,x). One can approximate (ii) as follows[5]:

$$y(n',s) = \sum_{m'=-\infty}^{\infty}\sum_{l'=-\infty}^{\infty} x'(m'+n',\ell')\,\partial(m',s\ell')$$
$$\approx \sum_{m'=-M'}^{M'}\sum_{l'=-L'}^{L'} x'(m'+n',\ell')\,\partial(m',s\ell') \quad \ldots.(iv)$$

where, $n' = \frac{-(N'-1)}{2}$ to $\frac{(N'-1)}{2}$, and …N = 2M + 1

In frequency domain, Discrete Radon transform is given by:

$$R\{x(m,l)\} = F^{-1}\{X_L(k)\Delta_{LS}(k)\}\ldots\ldots(v)$$

Where, $x(m,l)$ is the causal input image sequence in a shifted coordinate system defined as $x(m,l)=x'(m-2M', l-L')$, this coordinate axis shift is useful for the equivalent frequency domain computation of the Radon transform [5], $X_L(k)$ is the column wise Discrete Fourier Transform (DFT) of $x(m,l)$ and $\Delta_{LS}(k)$ is the DFT of the line δ function (shown in the fig.2) w.r.t. 'm' and is given in eqn.(vii). This line δ function is used to pick up the pixel intensities along different possible straight lines through the image. $X_L(k)$ is a row vector of support [0:L-1] and with the l'th element equal to $x(m,l)$, $\Delta_{LS}(k)$ is a matrix of support [0:S-1, 0:L-1] and S stands for the slope.

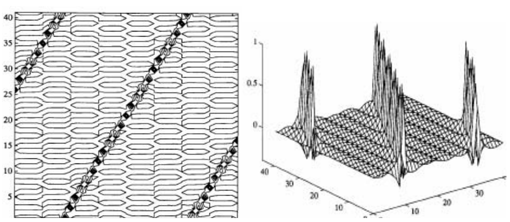


Fig.3 Line δ function for angle 60.95^o showing aliasing.

Inverse R-T is given by:

$$x(m,l) = F^{-1}\{W_S(k)\Delta_{SL}(k) \ldots\ldots\ldots(vi)$$

where, $W_S(k)$ is the filtered Radon transform corresponds to every row of the Radon transform passing through a filter of frequency response |ω| and $\Delta_{SL}(k)$ is given in eqn. (viii).

$\Delta_{LS}(k)$ and $\Delta_{SL}(k)$ are given by:

$$\Delta_{LS}(k) = \exp(-i2\frac{\pi k}{N}[\frac{N-1}{2} + g(s)(\ell - \frac{L-1}{2})]),\ \textit{for } 0 \le k \le N/2\text{-}d \quad \ldots\ldots(vii)$$
$$= \exp(i2\frac{\pi k}{N}[\frac{N-1}{2} + g(s)(\ell - \frac{L-1}{2})]),\ \textit{for } N/2\text{-}d+1 \le k \le N\text{-}1$$

$$\Delta_{SL}(k) = h(s)\exp(-i2\frac{\pi k}{N}[\frac{N-1}{2} + g(s)(\ell - \frac{L-1}{2})]),\ \textit{for } 0 \le k \le N/2\text{-}d \quad \ldots\ldots(viii)$$
$$= h(s)\exp(i2\frac{\pi k}{N}[\frac{N-1}{2} + g(s)(\ell - \frac{L-1}{2})]),\ \textit{for} (N/2\text{-}d+1) \le k \le N\text{-}1$$

where, N=length of columns of the image after zero padding,
L=length of the rows without zero padding,
l=0,1,2,……L-1, grid points along the row.
S=total number of scanning slopes.
g(s) determines the slopes of the line δ function for s=0,1,2,……S-1,

for linear slope sampling:

$$g(s) = B\frac{2s-S+1}{S-1} \quad \ldots\ldots \quad \ldots\ldots \quad \ldots\ldots\ldots (ix)$$
$$h(s) = -1 \ldots\ldots \quad \ldots\ldots\ldots \quad \ldots\ldots \quad \ldots \ldots\ldots(x) \qquad 0<B<\infty$$

for linear angular sampling:

$$g(s) = \tan(\frac{2s-S+1}{S-1}\arctan[\ B]), \quad \ldots..(xi)$$
$$h(s) = \sec^2(\frac{2s-S+1}{S-1}\arctan[\ B]), \quad \ldots..(xii) \qquad 0<B<\infty$$

Although this frequency domain technique is very efficient, there are two important periodicity constraints, imposed by Discrete Fourier Transform which are:
(i) line δ function aliasing for the slope, s>r as shown in the fig.3 and (ii) approximation of linear convolutions by circular (periodic) convolutions is not valid for the slope, s>r, where, image aspect ratio, r=length of the image /width of the image.

Another important problem is poor dispersion characteristics of the line δ function for higher slope. For a square image, r=1, hence slope of the line δ function is also limited to, s≤1 (i.e. max(θ)=45^o).

Kelley-Madisetti [5] solved these problems by padding zeros along the length of a square image to increase the aspect ratio, r of the image and using a new line δ function (known as DKMD formula) for the slope, s>1, which shows superior dispersion characteristic for higher slope. Disadvantage of this technique are that the number of computations which increases greatly because of zero padding and depending on the slope value there are three different flows of computations, which increases the computation complexity of this algorithm.

To overcome the problem of the original technique to transform the image features which are above ±45^o a square image space is virtually divided into two parts (fig.4) (i) image space below ±45^o (nonshaded part) and (ii) image space above ±45^o (shaded part). Now for the image features below ±45^o, X and Y axes remain unchanged, but for the image features above ±45^o, the coordinate axes have been inverted. Then clearly the features which were initially above ±45^o (w.r.t. X axes) are now within ±45^o. The Fast Radon Transform (FRT) and Inverse Fast Radon Transform (IFRT) of these two parts are done separately. For the IFRT, all the image features below ±45^o are developed with unchanged coordinate axes and referred as column wise operation where as all the image features above ±45^o are developed with inverted coordinate axes which is referred as row wise operation, then two IFRT's are added to develop the complete image. Hence to cover all the features of a square image there is no need to go for the slope, s>±45^o. In this way it avoids zero padding and requirement of any new line δ function. It results a highly regular algorithm with a significant reduction in number of computations and is given by a flow diagram in the fig.5. The symmetry between the forward and inverse algorithm in fig.5 are readily apparent. The proposed IFRT represents a theoretically sound method of inversion which avoids the need for data interpolation, and requires no classical back-projection operation and it is computationally as efficient as the forward FRT.

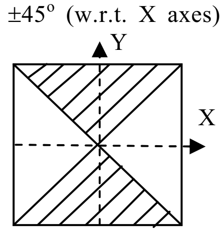


Fig.4 A square image space divided into two parts.

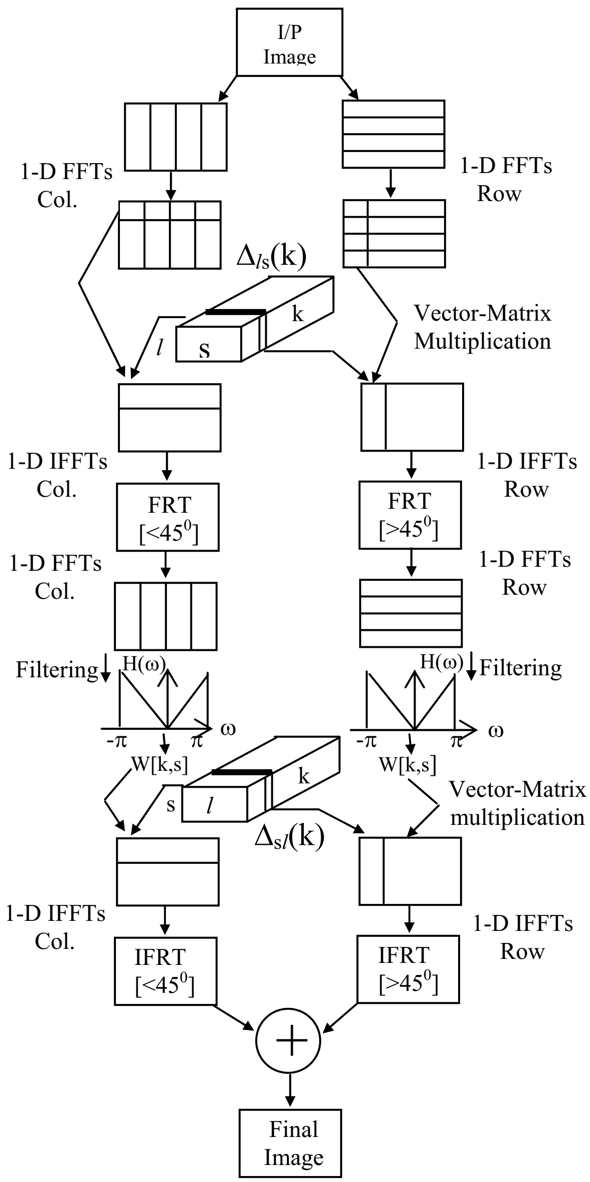


Fig.5 A flow diagram of the proposed FRT and IFRT algorithm.

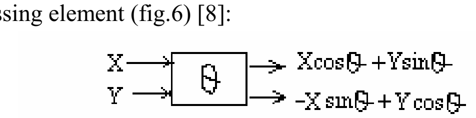
As shown in fig.5 the main computational steps are 1-D FFTs, Vector-Matrix multiplication and 1-D IFFTs which are discussed below:

(i) FFT computation

Using decimation in time algorithm, the basic operation of a M point FFT is: $a' = a + be^{-j(2\pi k/M)}$

$b' = a - be^{-j(2\pi k/M)}$

Where a′, b′, a, b are complex numbers. FFT computation may be done using butterfly structure using CORDIC as a basic processing element (fig.6) [8]:



A CORDIC unit

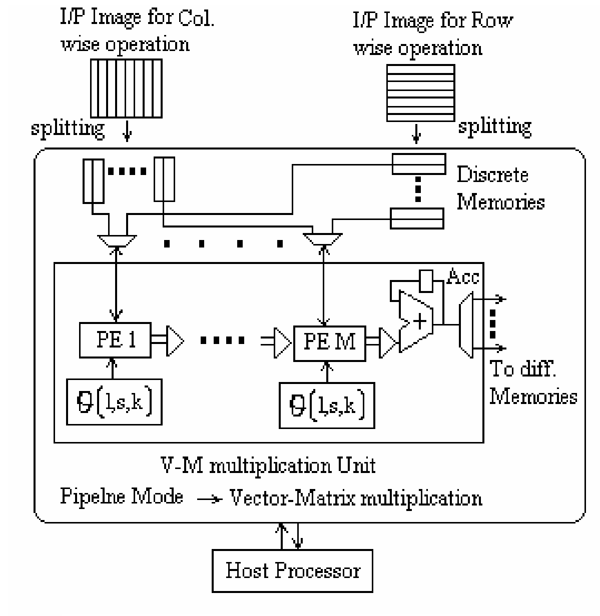


Fig.7a Proposed Hardware Architecture to realize vector-matrix multiplication.

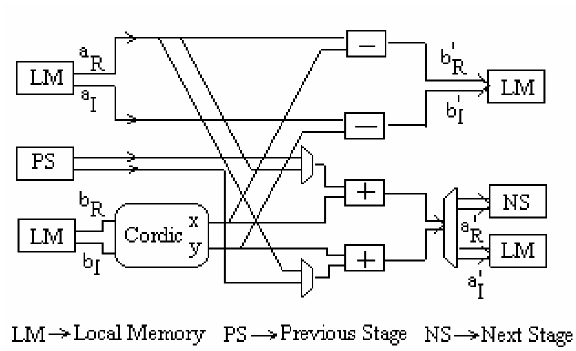


Fig.7b Processing element (PE)

Assuming our image to be processed is of size 256×256. As a realistic approach if p number of processing elements (PE) are used (p<256), total image is to be divided into p number of discrete memory banks, called Local Memory (LM) each containing 256/p number of columns. By accessing its LM, each PE computes 1-d FFT. Results may also be stored in the LM.
Each LM comprises of two independent memory banks for storing real and imaginary parts of the data.

(ii) Vector-Matrix (V-M) multiplication

$$[a_1 \quad a_2]\begin{bmatrix}\exp(-i\theta_{11}) & \exp(-i\theta_{12})\\ \exp(-i\theta_{21}) & \exp(-i\theta_{22})\end{bmatrix} = [b_1 \quad b_2]$$

CORDIC based PE shown in fig.7b may be used for V-M multiplication and corresponding input to the PE will be θ(l,s,k). Since the exponentials of the forward and inverse transform matrices are known explicitly [eqn. (vii) and (viii)] transform coefficients (θ's) can be generated in real time without storing it in a memory. During V-M computation PE's may be connected in pipeline as shown in fig.7a.

(iii) IFFT computation is done in a similar way as the FFT computation.

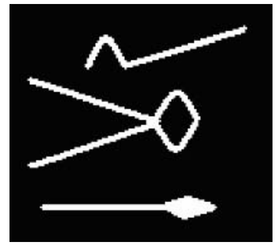
After completion of column wise operation PE may start computation for the row wise data. The host processor has now time to read processed data from the column memories. The host can then further process (high level processing) those data. For an input image, processed data of the two operations (row wise and column wise) corresponding to a particular co ordinate (x, y) may be added to develop a complete image, host processor can also be used for this purpose. Same architecture can be used for FRT and IFRT computations.

III. RESULT AND DISCUSSIONS

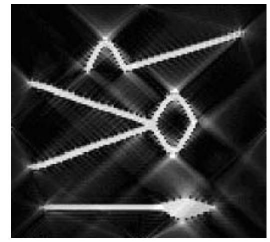
To verify our algorithm, an example (using MATLAB simulator) is shown in the fig. 8 with a 128×128 image, which illustrates the validity of the proposed algorithm. In the Kelly-Madisetti algorithm [5], depending on the slope value s there are three different flows of computations with complexities:
Number of computation=$O(N^3)$, when $s \leq 1$.
Number of computation=$O(N^3 \log N)$, when $1 < s \leq r$
Number of computation=$O(N^4)$, when $s > r$.
where, N is the zero padded length of the image and N>>L (L=width of the image), r is the aspect ratio of the image. In the proposed technique being max(s)=1, there is only a single flow of computation with the number of computations equal to $O(N^3)$, moreover N can be made equal to L without zero padding. Therefore a significant reduction in the number of computation is mainly because of using the most simple flow [$O(N^3)$] of the original technique and reduction of N itself to its minimum value without zero padding. However, for better resolution, image columns (or rows) are zero padded to double of its original length. In order to suppress the high frequency noise associated with the Ram-Lak filter, Shepp-Logan filter [1] was used in this reconstruction example. From the discussions it becomes clear that the complexity of the proposed algorithm is also much less compared to its previous algorithm because it requires only one type of flow instead of three different types of flows, which makes it possible to propose an efficient VLSI parallel processing architecture to implement the algorithm.

IV. CONCLUSION

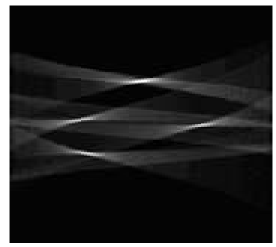
An Interpolation free, highly regular algorithm for the Fast Radon Transform [FRT] and Inverse Fast Radon Transform [IFRT] has been proposed. The computational complexities in the proposed method is much less compared to the existing technique results in a significant reduction in number of computations while preserving most of the important flexibilities of the original technique such as parallelism, ability to take any arbitrary set of projection angles etc. The high regularity and natural concurrency of the proposed algorithm makes it a suitable choice for VLSI implementation. Furthermore same architecture can be used for real time FRT and IFRT computations.



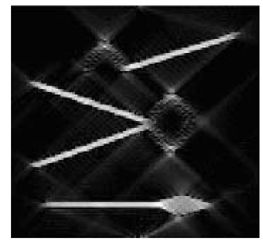
8a. Original Image.



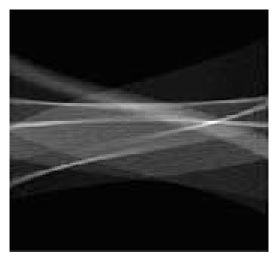
8b. Complete Reconstructed Image (fig.8d+fig.8f).



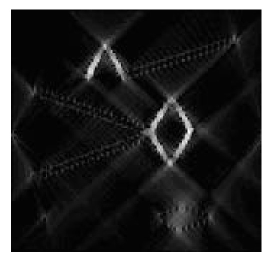
8c. RT for the features below ± 45°.



8d. IRT for the features below ±45°.



8e. RT for the features above ±45°.



8f. IRT for the features above ±45°.

Fig. 8 An example

References:

[1]. A.K. Jain, "Fundamental of Digital Image Processing", Englewood Cliffs, NJ: Prentice-Hall, 1988.
[2]. S.R. Deans, "The Radon Transform and Some of its Applications", New York: John Wiley and Sons, 1983.
[3]. P. Hurst, K.W. Current, I. Agi, E. Shieh, "A VLSI Architecture for 2-D Radon Transform Computations", IEEE Trans. ASSP, Vol.ASSP-2,pp. 933-936, April 1990.
[4]. G. Beylkin, "Discrete Radon Transform", IEEE Trans. ASSP, Vol.ASSP-35, No.2, pp. 162-172, Feb.1987.
[5]. B.T. Kelly, V.K. Madisetti, "The Fast Discrete Radon Transform-I: Theory", IEE Trans. Image Processing, Vol.2, No.3, pp. 382-400, July 1993.
[6]. B.T. Kelly, V.K. Madisetti, "The Fast Discrete Radon Transform", IEEE Trans. ASSP, Vol.ASSP-3, pp.409-412, March, 1992.
[7]. D. Lun, T.C. Hsung, W.C. Siu, "On the Convolution Property of a New Discrete Radon Transform and its Efficient Inversion Algorithm", IEEE International Symposium on Circuits & System, Vol.3, pp. 1892-1895, April 1995.
[8]. A. Banerjee et al., "FPGA realization of a CORDIC based FFT processor for biomedical signal processing", Microprocessor & Microsystems, Vol. 25/3, pp. 131-142, May, 2001.