

Image Reconstruction from Projections using Triangular Linear System of Equations

Johnes Jose

Department of Computer Science and
Engineering,
National Institute of Technology Calicut, India

V K Govindan

Department of Computer Science and
Engineering,
National Institute of Technology Calicut, India

Abstract

*In computed tomography, projections of an image are used to reconstruct that image. In this paper, a new method, which makes use of non-iterative algebraic approach, is proposed. The basic idea is to peel off the values of each pixel directly from the projections one by one. Projection angles and the ray width for peeling are determined. Using these parameters, a system of consistent triangular linear equations of the form, $Ax=B$ is formed. For a machine that creates an image of fixed size, the coefficient matrix A is pre-computed and stored in memory. Projection data provides the matrix B . Triangular system of linear equation is solved in linear time to get the individual pixel value. For an image of size $M \times N$, the proposed approach requires only M projection angles and N projections per view angle which is less when compared to existing methods. For an image of P pixels, the memory required is less than $2 * P^{3/2}$ which is better than the classical ART requirement of P^2 .*

1. Introduction

Computed tomography is the technique of reconstructing an image from its projections. It consists of passing a series of rays (in parallel, fan formation) through an object and measuring the attenuation in these rays by placing a series of detectors on the receiving side of the object. This data can be collected from various angles and the width of rays can be tuned. This technique finds uses in several applications such

as medical imaging, non-destructive testing evaluation, astronomy, material structure analysis and others.

1.1. Literature Survey

There are mainly two approaches to reconstruct an image from its projections. In the first technique, concept of Fourier slice theorem techniques is used. This technique requires the transformation of projections into frequency domain, filtering, and back projecting. Several such methods are present in the literature [1]. Gylson and Govindan [2] proposed a method to back project efficiently using Walsh Transform. These methods come under the category of transform approaches. In the second type of method, reconstruction is implemented directly in the signal space. This approach solves the problem using system of linear equations. In algebraic technique, various ideas have been proposed which can be broadly classified as iterative and non-iterative method. The accuracy of algebraic technique is superior. The main disadvantage of iterative algebraic reconstruction technique is their slow reconstruction speed and a large number of computations are required to for obtaining good results. Techniques such as Algebraic Reconstruction Technique (ART), its variants are iterative in nature. These method starts from an initial set of values which are tuned iteratively in such a manner that the computed projection data converge to available projection data. Kaczmarz [3] originally proposed ART method and used by Herman [4], Gordon [5] in image processing. These algorithms are flexible and allow applying different prior information about the object and this is of great importance when

we have incomplete projection data. The symmetric ART [6] is a variation of ART in which we perform one cycle of ART and in the next cycle equations are taken in the reverse order. The convergence rate of algebraic iterative algorithms is slow and lots of iteration should be made to obtain a good image construction. It is more efficient to apply these algorithms in parallel.

Kesidis and Papamarkos [7] proposed a method for reconstruction of grey scale images that uses algebraic techniques which is non-iterative in nature. It uses a technique called peeling. The projection samples are stored in an array and it contains an accumulation of sinusoidal contributions each one corresponding to certain pixel in the original image. The method proposes conditions on the angle of projection and width of the projection rays. This is to make sure that there is at least one sample in the accumulator array where only a pixel contributes. A decomposition sequence is also defined which provide the order in which samples are examined. For each identified pixel, its sinusoidal contribution is removed from the accumulator array. This process reconstructs the image from the outside inwards until the accumulator array is empty. The new method proposed in the paper comes under the non-iterative method.

1.2. Contribution

For an image of size $M \times N$, the newly proposed method requires only M projection angles and N projection samples per view angle. This is fewer than the method proposed by Kesidis and Papamarkos [7], in which an image of size 64×64 , requires 104 view angles and 1197 projection samples per view. Thus the imaging body suffers from fewer radiations. Proposed method can be parallelized and is computationally faster than [7]. For an image of P pixels, the memory required for the proposed approach is less than $2 * P^{3/2}$ which is better than the classical ART requirements of P^2 .

The rest of this paper is organized as follows: Section 2 provides basic idea, principles and various parameters involved in the proposed method. Section 3 provides performance comparison of the proposed approach with other methods. Finally, Section 4 draws the conclusions.

2. Proposed Approach

2.1. Basic Idea

To represent an image for reconstruction, we superimpose the image on an $M \times N$ grid. We assume that the value in a square grid is constant (pixel value). We also assume that the side of the grid is of unit length. The basic idea is to measure the value of each pixel one by one through a method called peeling. The method can compute pixel values either column wise or row wise. In this paper, column wise computation approach is used. The computation can be started either from left most or right most columns. Computing from right most columns is presented here. So in a grid of $M \times N$ pixels, the pixels are computed in the following order: (1, N), (2, N), (3, N) ..., (M, N), (1, N-1), (2, N-1) ..., (M, N-1), (1, 1), (2, 1) ..., (M, 1).

In order to be able to peel the values of a pixel in the above manner, the angle in which rays are projected and width of a ray are to be determined. Once this is done, we need to convert the above problem into a system of consistent linear equations which has $M \times N$ unknowns and $M \times N$ linear equations. Each projection constitutes a linear equation. Making the linear equations consistent is of prime importance.

2.2. Computation of Projection Angle

In order to compute the values of a pixel from projections, the ray should pass either through one pixel or if it passes through, say k pixels, then values of other $k-1$ pixels should be known. This means that each projection should not pass through more than one unknown pixel. This means while computing an unknown in an equation, all other variables which have a non-zero coefficient have to be known.

The idea is to have one set of parallel rays projected at a certain angle for determining the pixels of a given row. So in a grid of M rows, rays need to be projected at M different angles. These M different angles vary from 0 degree to 45 degree. Ray projected at 45 degree is used for computing the values of pixels in the first row. Projection angle moves towards to 0 degree for computing the pixels of the lower rows of the image grid. The angle in which the ray needs to be projected can be computed as follows:

For i^{th} row, projection angle will be $90 - \arctan(i)$. (1)

The angle of projection decreases as we move from row

i to row $i+1$. Figure 1 shows the projection angle for computing the pixels for row 3.

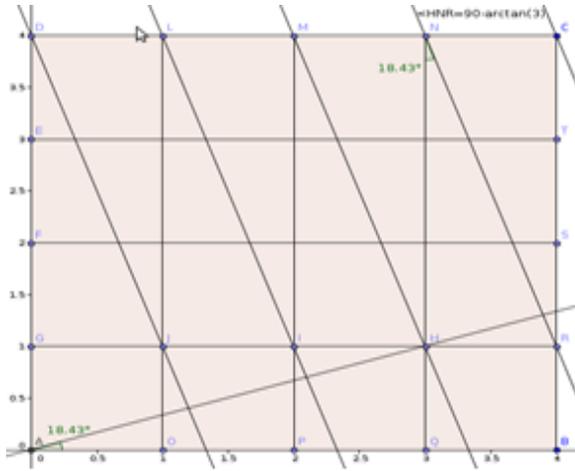


Figure1: Projection angle for computing pixels of row 3

2.3. Computation of Ray Width

As stated before, the whole idea of determining the projection angles and ray width is to make sure that we will be able to generate a system of consistent linear equations with unique solutions. So in an image of size $M \times N$, there should be $M \times N$ projections that could determine the values of each $M \times N$ pixels. Adjusting the width of rays along with the previously mentioned projection angles would enable us to achieve these criteria. There are two ways through which we could achieve these criteria. In the first method, the ray width is varied for each projection angle. In the second method, the ray width is held constant for all angles but the spacing between the parallel ray beams of a projection is adjusted.

2.3.1. Ray width varies for each projection angle

Each projection ray is intended for determining the value of a specific pixel. In order for the ray projected at angle θ not to pass through more than one unknown pixel, the width of the ray must not be greater than $\cos \theta$. A value less than this will not provide any benefit and would only make the computations complex. So the width of a ray projected at angle θ is given by the relation: $\cos \theta$. Width of the ray increases as angle of projection decreases.

2.3.1. Ray width fixed for all projection angles

In the next approach, the width of the ray is made constant for all projection angles. As mentioned in the previous case, the ray should not go through more than one unknown pixel. This implies that width should be the minimum of width computed for different projection angles. This turns out to be the width of ray projected for first row. The angle of projection for first row is $\arctan(1) = 45$ degree, and hence the ray width would be $\cos 45$. A value less than this will not provide any benefit and would only make the computations complex.

For an image of size $M \times N$, the number of parallel ray beams in a projections will be N . By fixing the width of ray to $\cos 45$, there should be a gap between consecutive parallel beam rays of a projection. For rays projected at angle θ , this gap should be $\cos \theta - \cos 45$.

2.4. Contribution of a Pixel to a Projection

The area contribution of a pixel to a particular ray of a projection can be pre-computed for an image of a particular size. The values can be efficiently stored in the memory so that during back projection, this data can be used for computing pixel values. This reduces the time required for image reconstruction. The area that a ray covers in a pixel can be computed based on area property of trapezium. Length of parallel sides depends upon the angle of projection θ . The perpendicular distance between the parallel sides is of unit length. For a ray beam projected at an angle θ intended to compute the value of a pixel at row R , the set of values computed by the equation: $1 - (0.5 + j) * \tan(\theta)$ (2), where j varies from 0 to $R-1$ encloses the area covered by projection on pixels.

2.5. Generating Triangular System of Linear Equations

Once the contribution of a pixel to a projection is determined, we can generate a system of triangular linear equations. We label the pixels from 1 to $M \times N$ column wise from left bottom to right top. For example, for an image of size 4 x 5, the pixels would be labeled as given in the Figure 2.

The rays are projected at 4 different angles and in each angle of projection, there would be 5 rays. These

20 rays form 20 linear equations. The angle of projection and width of ray determined previously allow them to be mapped into a triangular system of linear equations. This can be formulated into a standard linear equation $Ax=B$, where A denotes what percentage of pixel area contributes to the particular projection, x the pixel values that is to be computed and B the projections.

	4	8	12	16	20
	3	7	11	15	19
	2	6	10	14	18
	1	5	9	13	17

Figure 2: Labeling of pixels for an image of size 4 x 5

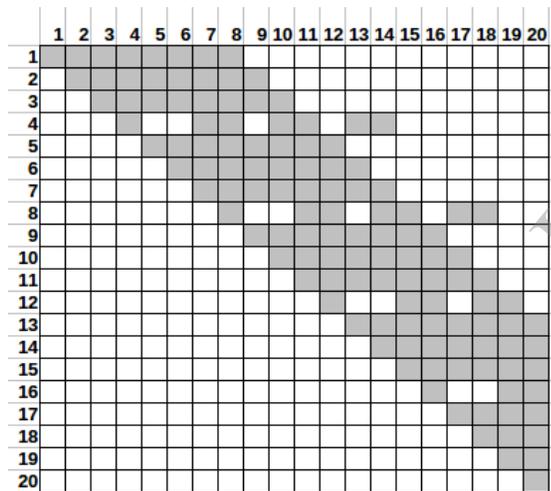


Figure 3: Coefficient matrix for an image of size 4 x 5

Figure 3 denotes coefficient matrix for an image of size 4 x 5. The horizontal rows denote the 20 linear equations (projections) and vertical columns denote the contribution of a pixel to a linear equation (projections). The shaded pixels indicate that it contributes to that projection. For example, for projection 5, pixels from 5 to 12 contribute. This means that in the coefficient matrix for row 5, coefficients in columns 5 to 12 are non-zero and rests are zero. This matrix is constant for an image of fixed size.

As stated before, equation (2) governs the set of values that constitute the areas covered on pixels by a ray projected at angle θ intended for computing the value of pixel i at row R . The format $pixel_num \{value\}$ denotes that the projection passes through pixel $pixel_num$ and its area contribution to the projection is $value$. Let $array$ contains the following values in that order $1 - (0.5 + j) * \tan(\theta)$, where j varies from 0 to $R-1$. Following pattern depicts the pixels that contribute to projection at angle θ intended for computing pixel i along with its area contribution to the projection.

$$(i+k)\{array\ 1^{st}\ element\}, (i+k+M)\{1-array\ 1^{st}\ element\},$$

$$(i+k-1)\{array\ 2^{nd}\ element\}, (i+k-1+M)\{1-array\ 2^{nd}\ element\},$$

$$\dots$$

$$(i)\{array\ R^{th}\ element\}, (i+M)\{1-array\ R^{th}\ element\},$$

$$(i+M-1)\{array\ 1^{st}\ element\}, (i+M-1+M)\{1-array\ 1^{st}\ element\},$$

$$(i+M-2)\{array\ 2^{nd}\ element\}, (i+M-2+M)\{1-array\ 2^{nd}\ element\},$$

$$\dots$$

$$(i+M-k)\{array\ R^{th}\ element\}, (i+M-k+M)\{1-array\ R^{th}\ element\},$$

$$\dots$$

Follow the pattern until conditions $pixel_num \leq M \times N$, $pixel_num \bmod M \neq 0$ holds, where

- $i + k$ is a multiple of M ,
- k is $M - pixel_num \bmod M$,
- M is the number of rows
- N is the number of columns

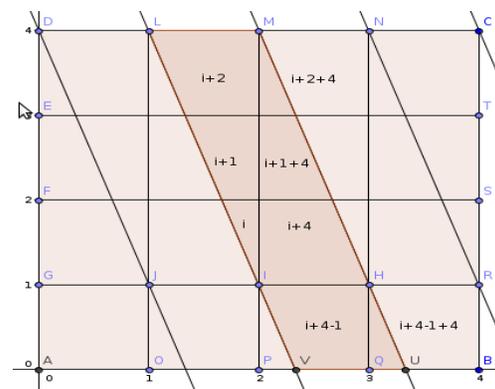


Figure 4: Pixels through which projection intended for computing value of i passes.

For a machine that reconstructs an image with a fixed size, its coefficient values can be computed using

the above pattern and store permanently in the memory. As it is already known, which all pixels contribute for a particular projection ray, only that information needs to be stored in memory. Triangular linear equations are generated using the above stored data and projection values. Solving the triangular system of equations is easy as lower equations contain only later variables. Rudnei [8] provided parallel solutions for triangular system of linear equation.

The whole procedure using proposed approach can be summarized as follows:

- For a machine that reconstructs an image with fixed size, the coefficient matrix values are constant and are efficiently stored in the memory.
- Rays are projected at an angle $90 - \arctan(i)$, i varies from 1 to number of rows.
- Width of the ray can either vary or be fixed with projection angle. For the first case, width of a ray projected at angle θ is $\cos \theta$. For the second case, width of ray is $\cos 45$ and there should be a gap between consecutive parallel rays of a projection. For rays projected at angle θ , this angle should be $\cos \theta - \cos 45$.
- Projection values are stored and used for constructing triangular system of linear equation $Ax=B$, where A represents the stored coefficient matrix, x presents the pixel values to be computed and B represents the computed projection values.
- Triangular linear equations are solved efficiently to get the pixel values. Parallel algorithm for solving triangular linear equation can be used to improve the speed.

3. Performance Comparison and Discussion

Let us consider a square image with $N \times N$ where N is the number of rows or columns in the image, so, the total number of pixels in the image is $P = N^2$. The total number of projection angles in the proposed method is N and there is N projection samples per view angle. In the method proposed by Kesidis and Papamarkos [7], in an image of size 64×64 , there would be 104 view angles and 1197 projection samples per view angle. The above result shows that the proposed approach requires fewer projection angles and fewer projection

samples per view angle. In medical imaging, this seems important as the body suffers from fewer radiations. As the projection angle ranges from 0 to 45 degree, this method finds useful when projections from limited angle is possible. The computational time required is considered as the sum of time required to compute the projection samples plus the image reconstruction time. The reconstruction time is lesser as the coefficient matrix is pre-computed and the algorithm is parallelized. The memory required for classical ART is given by the relation $Mem_{ART} = P^2$. The memory required for the proposed method is given by the relation $Mem_P \leq 2 * P^{3/2}$.

4. Conclusion

This paper proposes a simple method for reconstructing an image in the signal space non-iteratively. Initially proper values for angle of projections and ray width are computed. This is a key criterion in constructing a system of consistent triangular linear equations $Ax=B$. For a machine that creates an image of fixed size, the coefficient matrix A is pre-computed and stored in memory. Projection data gives the value of B . Triangular system of linear equation is solved in linear time to get the individual pixel value. The algorithm is amenable for parallel implementation, which drastically reduces the reconstruction time. The performance of the approach is found to be better than classical ART and the recent approach proposed by Kesidis and Papamarkos [7]. Future work includes investigating further techniques that increase the speed of computation and reduce the memory required.

5. References

- [1] A. C. Kak, and Malcolm Slaney, "Principles of Computerized Tomographic Imaging", IEEE Press, 1988.
- [2] Gylson Thomas, and V. K. Govindan, "Computationally efficient filtered-back projection algorithm for tomographic image reconstruction using Walsh transform", Journal of Visual Communication and Image Representation", 17(3), June 2006.
- [3] Kaczmarz S, "Angenäherte Auflösung von Systemen linearer Gleichungen", Bulletin International de l'Académie Polonaise des Sciences et des Lettres, 35, 1937, 355–357.

- [4] G. T. Herman, "Image reconstruction from projections: The fundamentals of computer tomography", Academic Press, 1980.
- [5] R. Gordon, "A tutorial on art (algebraic reconstruction techniques)", IEEE Transactions on Nuclear Science, 21(3), 1974, 78–93.
- [6] A. Bjorck, and T. Elfving, "Accelerated projection methods for computing pseudoinverse solutions of linear equations", BIT, 19(2), 1979, 145–163.
- [7] A. L. Kesidis, and N. Papamarkos, "Exact image reconstruction from a limited number of projections", Journal of Visual Communication and Image Representation, 19(5), July 2008, 285–298.
- [8] Rudnei Dias, and Tim Hopkins, "The parallel solution of triangular systems of linear equations", Proceedings of Second Symposium in High Performance Computing, Amsterdam, October 1991, 245-256.

IJERT