Implementation of efficient image reconstruction for CT

Jie Liu (刘 杰) and Guangfei Wang (王光飞)

Department of Biomedical Engineering, school of computer, Beijing Jiaotong University, Beijing 100044

Received November 19, 2004

The operational procedures for efficiently reconstructing the two-dimensional image of a body by the filtered back projection are described in this paper. The projections are interpolated for four times of original projection by zero-padding the original projection in frequency-domain and then inverse fast Fourier transform (FFT) is taken to improve accuracy. Nearest interpolation is applied to decrease operation time. Projection dependence of next pixel at the same row is used. For each row of image, the first pixel projection once for each angle is pre-computed, other pixel projection of this row can be found by iteration. Therefore, the pre-interpolation process only has to be performed once for each row, rather than individually for each pixel. It greatly reduces the amount of computation. Compared with original implementation, the speed of reconstruction image is nearly improved by five times. The image accuracy is still preserved.

OCIS codes: 100.3010, 100.6950.

Computer Tomography (CT) refers to the cross-sectional imaging of an object. When a series of rays pass through an object, the ray attenuation occurs. After projections are collected by many detectors on the side of the object, image reconstruction algorithm can be used to reconstruct its two-dimensional (2D) cross-sectional image from its projection[1]. The mathematical basis for tomography was inverse radon transform theory. The filtered back projection algorithm is widely used by almost all commercially CT. Compared with algebraic techniques and maximum entropy method for reconstruction, the filtered back projection method is much superior to other methods[2]. Although the principle of filtered back projection is simple, general implementation of the filtered back projection algorithm is time consuming. There are some fast reconstruction algorithms[3−5]. These algorithms generally compute back projection by increment and decomposition so that the result is not optimized. This paper discuses the efficient implementation of the filtered back projection algorithm for reducing computing time. In our implementation, first, the filtered projections for a very large number of points are found by zero-padding the original projection and nearest interpolation is selected to increase accuracy and speed. Second, projection dependence of next pixel for each row is applied to only computer the first pixel projection of each row by pre-interpolation process once. The computation of other pixels in this row only involves one addition and one multiply. Therefore, it greatly reduces the amount of computation.

The geometry of the back projection process at angle θ is illustrated in Fig. 1. First, two kinds of coordinate system are selected. The image spatial coordinates are (x, y). The origin of this spatial coordinate system is located at the center of the image. The image pixel coordinates are (i, j) and the origin of this coordinate system is located at the left corner of the image. Second, we express some parameters, Δx and Δy are defined as pixel space along x and y dimensions. δ is space between neighbor projection. L is spatial length of projection vector. $M \times N$ is image array. $R_0[j]$ is the distance of the projection for the first pixel in row j of the image. The relations between the image spatial coordinates and the image pixel coordinates are simply given by

$$x = [i - (N-1)/2]\Delta x, \quad y = [j - (M-1)/2]\Delta y,$$

$$i = 0, 1, 2, \cdots, (N-1); \quad j = 0, 1, 2, \cdots, (M-1).$$

The projection at angle θ is represented by $P_\theta[k]$ of discrete projection values. k is the index from the left end of projection line. δr is the element space between values of $P_\theta[k]$. The different projection distances are expressed as

$$r = x\cos\theta + y\sin\theta = k\delta r - L/2, \quad R = L/2 + r = k\delta r,$$

$$k = 0, 1, \cdots, (K-1),$$

$$R[i, j] = L/2 + (i - (N-1)/2)\Delta x\cos\theta$$

$$+[j - (M-1)/2]\Delta y\sin\theta,$$

$$i = 0, 1, 2, \cdots, (N-1); \quad j = 0, 1, 2, \cdots, (M-1),$$

where r is distance relative to the center of the projection line and $L/2$ is the distance from the beginning of

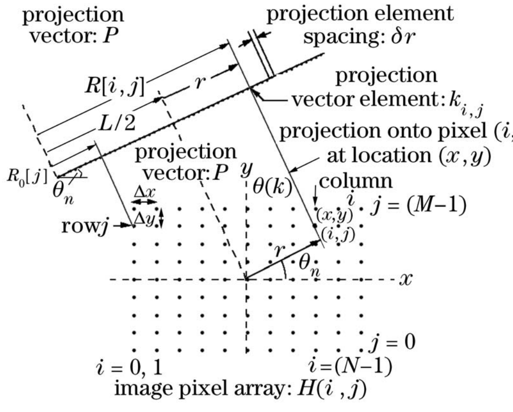

Fig. 1. Back projection.

the line to location of center projection (L is the spatial length of projection vector). R is the corresponding distance from the left end of the projection line and it is corresponding to element $P_\theta[k]$ of the projection. $R[i,j]$ is the location of the projection for the (i,j) image pixel on the given projection line. Generally, $R \neq R[i,j]$. Therefore, given value of $R[i,j]$ for the projection of the (i,j) image pixel, we can compute the corresponding fractional index value $k = R[i,j]/\delta r$ and interpolate the projection to obtain the projection $P_\theta(R[i,j])$. The final value of each pixel will be the result of the accumulation of all projection values at that pixel. Therefore, the key of back projection is how to compute $R[i,j]$ efficiently:

$$I(i,j) = \sum_{\theta=0}^{\pi} P_\theta(R[i,j]) = \sum_{n=0}^{N-1} P_n(R[i,j]),$$

$$i = 0,1,2,\cdots,(N-1), \quad j = 0,1,2,\cdots,(M-1). \quad (1)$$

A frequency-ramp filter should be used to raw projection. In order to increase accuracy, we need interpolate much data. The sample space δr fixed by the physics of the projection measurement system is taken. However, the interpolation space $\delta r'$ can be as small as one like. The cost of interpolation is computation and extra memory requirement. In fact, if spatial-domain interpolation is done directly, it is quite sufficient. Nevertheless, zero-padding the Fourier transform of the projection and then taking inverse transform of this zero-padded result can do spatial-domain interpolation. If larger point sizes are used, zero padding has better accuracy. Because frequency-ramp filtering process is usually performed using fast Fourier transform (FFT), projection pre-interpolated is computationally efficient to combine these two processes. As zero padding a data sequence yields more data points in the frequency domain. The measured value can be pre-interpolated to a much finer uniform spacing. The cost of interpolation is computation and extra memory requirement. It was shown that there is little advantage to using an interpolation ratio which is much greater than 8. Generally, the ratio is 4. It means that interpolation projection is 4 times of measured projection. By this way, we increase the data accuracy. Compared to linear interpolation in spatial domain, we also improve the speed of computation. It was also shown that zero-padding the data to four times its length also can remove the dishing and disc artifacts from the final images. This process is follow:

1) Zero padding the original projection from K' points to the next higher power of 2, $N_1 = 2^k$;

2) Taking the N_1 points real FFT, this result is $V'_\theta(k)$;

3) Applying the frequency ramp filter: $|k|V'_\theta(k)$;

4) Interpolating $|k|V'_\theta(k)$ from N_1 points to $N_2 = LN_1$, interpolation factor $L = 4$;

5) Taking N_2 point inverse real FFT of the zero-padded sequence, the original sequence has been pre-interpolated or over-sampled by a factor L.

In Eq. (1), the computation includes three-process, $R[i,j]$, interpolation, and back projection. Back projection consumes the maximum time (about 70%). Therefore, it is essential to find efficient implementation technique. An efficient implementation of the back projection algorithm bases on the fact that the parameters such as Δx, Δy, δ, L, M, and N all have fixed values. They are constant for all projections. The angle is also a constant for each incidence of the back projection process. Therefore, terms that use only these parameters can be pre-computed initially. The location on the projection line of the projection of the (i,j) pixel can be rewritten so that the projection $R[i,j]$ in the image is computed as

$$R[i,j] = R_0[j] + i\Delta r, \quad \Delta r = \Delta x \cos\theta_n,$$

$$R_0[j] = L/2 + [j-(M-1)/2]\Delta y \sin\theta_n$$
$$-[(N-1)/2]\Delta x \cos\theta_n,$$

$$i = 0,1,2,\cdots,(N-1), \quad j = 0,1,2,\cdots,(M-1),$$

$$\theta_n, n = 1,2,\cdots,N_p, 0 < \theta_n < \pi.$$

$R_0[j]$ is the distance of the first pixel in row j of the image pixel array. $R_0[j]$ is key for back projection. As stated above, $R[i,j]$ consists of $R_0[j]$ and Δr. $R_0[j]$ is only related with j. For given j, $R_0[j]$ is need only be pre-computed once for each angle θ once. Δr is offset of column i. Therefore, the computation of $R[i,j]$ for other column at this row only involves one addition and one multiply for each of the $M \times N$ pixels in the image. Generally, the values $R[i,j]$ do not correspond exactly to the values R that are associated with the elements of the projection so that interpolation of the projection required to obtain the desired projection value. However, this computation of interpolation process is burdensome. This process can be solved by pre-interpolating the projection and then used a nearest-neighbor look-up table of the pre-interpolated projection to obtain the desired value $P_\theta(R[i,j])$. This pre-interpolation process only has to be performed once for each projection, rather than individually for each pixel. Therefore, it greatly reduces the amount of computation. Given $R[i,j]$, the nearest-neighbor value of $P_\theta(R[i,j])$ is obtained by look-up table from $P_\theta(k)$. The fractal index k'_{ij} is corresponding to $R[i,j]$. k_{ij} is the nearest integer value of k'_{ij}: $k_{ij} = \text{int}[k'_{ij}] = \text{int}[S_0[j] + i\Delta s]$; $S_0[j] = R_0[j]/\delta r = [L/2 + [j-(M-1)/2]\Delta y \sin\theta - [(N-1)/2]\Delta x \cos\theta]/\delta r$; $\Delta s = \Delta r/\delta r = \Delta x \cos\theta/\delta r$, $i = 0,1,2,\cdots,N-1$, $j = 0,1,2,\cdots,M-1$. $S_0[j]$ is the fractional index of projection corresponding to the projection of the first pixel in row j. $\text{Int}[S_0[j]]$ means the integer nearest to the fractional value of $S_0[j]$. Δs is the fractional projection index space between the projection of two adjacent image pixel in the same row of the image. For each row of image, We only pre-computed the first pixel once for each angle, Other pixels of this row can be found by iteration. It greatly reduces the amount of computation. Compared with original implementation, the speed of reconstruction image is nearly improved by five times. All interpolated projection of each angle is accumulated to final image.

In order to compare different methods, we reconstruct image of 256 × 256 pixels illustrated in Fig. 2. Figure 2(a) is original image. In Fig. 2(b), we see the reconstructed image in our efficient algorithm. First, we pre-interpolate the projection to four times original length by zero-padding in FFT and use nearest look up table to obtain the filtered projection. Second, our efficient back projection is applied to reconstruct image. We use next

Table 1. CPU Time for Reconstructing 256 × 256 Image on Pentium 1G

Total Time (s)	Interpolation	Time (s)	Back Projection	Time (s)	PSNR (dB)
1.02	Nearest Interpolation	0.21	Efficient	0.81	65
5.11	Linear Interpolation	0.68	General	4.43	61

Fig. 2. Reconstruction image: (a) original image; (b) zero-padding interpolation; (c) general method.

pixel dependence for each row and only pre-computer the first pixel projection for given row. Other pixel projection can be iterated. Therefore, it reduces operation time greatly. We also use general linear interpolation without zero padding and direct back projection. The reconstructed image is illustrated in Fig. 2(c). We use peak signal noise ratio (PSNR) as image comparison criterion[6]. The result is listed in Table 1. The result shows that Fig. 2(b) reconstructed by our efficient implementation is better than Fig. 2(c) which was reconstructed by general method. Furthermore, our algorithm time is only about 20 percentage of general method for 256 × 256 pixels image. Table 1 illustrates the time of different method.

The filtered back projection is critical part in CT Imaging. Fast algorithm and excellent quality are very important. In this paper, we introduce the filtered back projection algorithm which has some merits. First, spatial-domain interpolation can be implemented by zero-padding the Fourier transform of the projection and then taking inverse transform of this zero-padded result to increase the accuracy of projection. It was also shown that zero-padding the data to four times its length can remove the dishing and disc artifacts from the final images. Second, We only pre-computed the first pixel projection at each row of image once for each angle, other pixel projection of this row can be found by iteration. Third, the nearest interpolation is taken by look up table to improve speed of operation. The process of image reconstruction is speeded up and the quality of image is still preserved.

J. Liu's e-mail address is lj@computer.njtu.edu.cn.

References

1. A. C. Kak and M. Slaney, *Principle of Computerized Tomographic Imaging* (IEEE Press, New York, 1988).
2. P. Zhang and Z. T. Zhang, Computerized Tomography Theory and Application (in Chinese) **10,** (4) 4 (2001).
3. I. Agi, P. J. Hurst, and K. W. Current, IEEE J. Solid State Circuits **28,** 210 (1993).
4. S. Basu and Y. Bresler, IEEE Trans. Med. Imaging **9,** 1760 (2000).
5. S. Mate and I. Bajla, IEEE Trans. Image Processing **9,** 421 (1990).
6. T. Zhang and J. B. Yan, Computerized Tomography Theory and Application (in Chinese) **9,** (1) 13 (2000).