CT의 투영을 위한 빠른 선적분 계산 방법

천권수[1], 길준민[2]
[1]대구가톨릭대학교 방사선학과
[2]대구가톨릭대학교 컴퓨터소프트에어학부
kschon@cu.ac.kr, jmgil@cu.ac.kr

Fast Calculation of Line Integral for Projection on CT Array

Kwon Su Chon[1], Joom-Min Gil[2]
[1]Dept. of Radiological Science, Daegu Catholic University
[2]School of Computer Software Engineering, Daegu Catholic University

요 약

CT의 반복재구성방법은 투영과 역투역을 번갈아가며 최적의 단면영상을 얻을 때까지 반복한다. 영상재구성 시간을 단축하기 위하여 시간이 많이 소요되는 투영을 빠르게 수행할 수 있는 알고리즘이 필요하다. 본 논문은 Siddon 알고리즘을 개선한 Jacobs 버전보다 대략 10% 빠른 알고리즘을 제안한다. 평행빔의 경우에 대해 조사되었지만 향후 부채살빔 및 콘빔의 경우로 확장이 가능하다.

1. 서론

CT(computed tomography)는 환자의 병변 진단을 위해 활발하게 활용되고 있다. 최근 방사선피폭에 의한 2차 피해 발생으로 환자의 방사선 피폭선량을 줄이기 위한 노력이 진행되고 있다. 이를 위해 환자에게 노출되는 엑스선(X-ray)을 줄이는 것이 필수적이다. 이 경우 CT 검출기에서 획득되는 투영영상에 잡음이 증가한다[1]. 잡음에 강한 특성을 갖는 반복재구성기법이 영상재구성방법으로 각광받고 있다. 반복재구성방법은 투영(projection)과 역투영(back-projection)을 반복하면서 최적의 단면 영상을 찾아가는 수치계산 방법으로 시간이 오래 걸리는 단점이 있다[2, 3]. 이러한 단점을 극복하기 위해 반복횟수를 줄이거나, 투영-역투영의 계산속도를 향상시키는 등 여러 방법이 시도되고 있다.

역투영은 투영의 전치행렬(tranpose matrix)로 표현되기 때문에 투영에서 빠른 속도가 보장되면 역투영도 빠르게 계산할 수 있다. 투영-역투영의 속도를 높이기 위해서는 투영에서 계산시간이 오래 소요되는 선적분 계산을 빠르게 수행할 수 있는 알고리즘이 요구된다. 선적분을 효과적으로 계산하는 방법이 Siddon에 의해 제안되었고[4], 현재 개선된 Siddon 방법(Jacobs 버전)[5]이 광범위하게 활용되고 있다. 본 연구는 투영의 속도를 더욱 빠르게 계산할 수 있는 개선된 새로운 알고리즘을 제안한다. 평행빔의 경우로 제한하여 알고리즘을 검증한다.

2. CT 어레이에 대한 광선의 선적분

CT에 사용되는 엑스선은 직선으로 묘사된다. 직선이 CT 단면 영상을 지나갈 때 픽셀과 만나는 선분의 길이와 그 픽셀값을 곱한 것을 직선이 지나는 모든 픽셀에 대해 누적하면 선적분(line integral)이 계산된다. $N_x \times N_y$ 크기를 갖는 CT 어레이(array)에서 하나의 광선에 대한 선적분은 다음과 같이 계산된다.

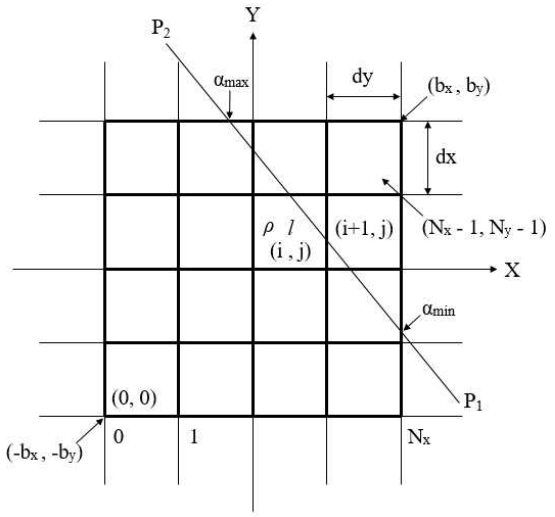
$$proj = \sum_{(i,j \subset l)} l(i,j)\rho(i,j) \quad (1)$$

여기서 *i*와 *j*는 영상의 폭과 높이 방향의 인덱스(index)이고 *l(i,j)*는 *(i,j)* 픽셀과 광선(직선)이 만나는 길이를 나타내고 *ρ(i,j)*는 그 픽셀값을 나타낸다. (그림 1)은 수학 좌표계와 영상을 동시에 나타낸 것이다. 점 $P_1(p_{1x}, p_{1y})$에서 출발한 엑스선은 $P_2(p_{2x}, p_{2y})$에서 검출되고 직선

$$p_x = p_{1x} + \alpha(p_{2x} - p_{1x}) \quad (2)$$

$$p_y = p_{1y} + \alpha(p_{2y} - p_{1y}) \quad (3)$$

로 표시된다. 여기서 *a*는 위치를 결정하는 직선 매개변수로 두 점 사이의 상대적인 거리를 나타낸다.



(그림 1) CT 어레이를 지나가는 광선

광원과 검출기를 나타내는 두 점은 CT 어레이 외부에 위치한다고 가정한다. 직선이 통과하는 CT 어레이를 (그림 1)과 같이 일정하게 배치된 X 및 Y축에 평행한 직선으로 구성된 그리드(grid)로 간주할 수 있다. X축에 수직인 그리드와 Y축에 수직인 그리드를

$$g_x = -b_x + i \times dx \tag{4}$$

$$g_y = -b_y + j \times dy \tag{5}$$

로 나타낼 수 있다. 여기서 i와 j는 그리드를 나타내는 인자로 i = 0, 1, ⋯, N_x와 j = 0, 1, ⋯, N_y의 범위를 갖는다. i와 j가 N_x-1와 N_y-1까지 사용될 경우 픽셀을 나타내는 인자로 사용될 수 있다. 광선을 나타내는 식(1)과 식(2)가 X축 및 Y축 그리드와 만날 때의 직선 매개변수는

$$\alpha_x(i) = \frac{(-b_x + i \times dx) - p_{1x}}{p_{2x} - p_{1x}} \tag{6}$$

$$\alpha_y(j) = \frac{(-b_y + j \times dy) - p_{1y}}{p_{2y} - p_{1y}} \tag{7}$$

가 된다. 여기서 $p_{2x} = p_{1x}$, $p_{2y} = p_{2y}$ 경우는 제외한다. 광선이 CT 그리드에 처음과 마지막으로 만나는 직선 매개변수 (α_{min}, α_{max})을 결정할 수 있고 이 때의 픽셀을 각각 (x_F, y_F)와 (x_L,y_L)라 하고 $\alpha_x \leftarrow \alpha_x(x_F + sign_x)$와 $\alpha_y \leftarrow \alpha_y(y_F + sign_y)$로 대입한다. 여기서 $p_{1x} < p_{2x}$이면 sign_x = 1이고, 반대이면 sign_x = 0이다. sign_y도 비슷하다. 그리고 X축 직선 매개변수의 증감인자를

$$\alpha_{xu}(i) = \frac{dx}{| p_{2x} - p_{1x} |} \tag{8}$$

라 하자. Y축 방향의 직선 매개변수의 증감인자도 비슷하게 표현된다. X 방향의 픽셀의 증감은

$$i_u = \begin{cases} 1, & p_{1x} < p_{2x} \\ -1, & p_{1x} > p_{2x} \end{cases} \tag{9}$$

로 나타낼 수 있다. Y 방향의 픽셀 증감도 비슷하게 표현된다.

광선과 CT 어레이가 만나는 최초의 직선 매개 변수를 $\alpha_c = \alpha_{min}$로, 픽셀을 $i = x_F$와 $j = y_F$로 초기화하고 $\alpha_x < \alpha_y$((그림 1)은 이 조건이 만족)의 검사를 통해 첫 번째 픽셀을 지나는 광선의 길이

$$l(i,j) = (\alpha_x - \alpha_c)d \tag{10}$$

를 계산할 수 있다. 여기서 d는 P_1과 P_2 사이의 거리이다. 픽셀을 지나는 선 길이와 픽셀값의 곱

$$l(i,j)\rho(i,j) \tag{11}$$

을 다음 픽셀에 대해 반복 계산하여 누적하면 선적분이 계산된다. 다음 픽셀에 대해 반복하기 전에

$$i \leftarrow i + i_u \tag{12}$$

$$\alpha_c \leftarrow \alpha_x \tag{13}$$

$$\alpha_x \leftarrow \alpha_x + \alpha_{xu} \tag{14}$$

와 같이 픽셀과 직선 매개변수를 업데이트 한다. 만약 $\alpha_x > \alpha_y$인 경우에는

$$j \leftarrow j + j_u \tag{15}$$

$$\alpha_c \leftarrow \alpha_y \tag{16}$$

$$\alpha_y \leftarrow \alpha_y + \alpha_{yu} \tag{17}$$

로 업데이트 한다.

픽셀값과 선분의 곱이 계산되는 루프의 횟수는

$$N = \begin{cases} x_L - x_F + 1, & | p_{2x} - p_{1x} | > | p_{2y} - p_{1y} | \\ y_L - y_F + 1, & | p_{2x} - p_{1x} | < | p_{2y} - p_{1y} | \end{cases} \tag{18}$$

와 같다. 루프의 최대 횟수는 N_x 또는 N_y를 넘지 않는다. 계산 알고리즘을 Algorithm 1에 유사코드로 나타내었다. 광선의 입사를 기울기를 기준으로 구별하였다. 이렇게 구별하면 X축 또는 Y축 방향의 반복 루프를 통해 선적분을 쉽게 계산할 수 있다. (그림 1)은 Y축 방향으로 반복 루프를 수행한. (i, j)로 표시된 픽셀과 (i+1, j) 픽셀이 한 번의 루프속에서 차례로 계산된다.

Algorithm 1 Fast Calculation Algorithm

```
Input: ρ[..], p1x, p2x, p1y, p2y, d, αmin, xF, yF,
       xL, yL, sign_x, sign_y, aux, auy, iu, ju
Output: proj
Initalize ac=αmin, i=xF, j=yF, sum=0, proj=0
          ax=ax(xF+sign_x), ay=ay(yF+sign_y)
if |p2x - p1x| > |p2y - p1y| then
   Initialize n=xF
   while n <= xL do
      if ay < ax then
         sum ← sum + (ay - ac) * ρ(i,j)
         j ← j + ju, ac ← ay, ay ← ay + auy
      end if
      sum ← sum + (ax - ac) * ρ(i,j)
      i ← i + iu, ac ← ax, ax ← ax + aux
   end while
else
   Initialize n=yF
   while n <= yL do
      if ax < ay then
         sum ← sum + (ax - ac) * ρ(i,j)
         i ← i + iu, ac ← ax, ax ← ax + aux
      end if
      sum ← sum + (ay - ac) * ρ(i,j)
      j ← j + ju, ac ← ay, ay ← ay + auy
   end while
end if
proj ← sum * d
```

3. 계산 결과

알고리즘의 성능은 선적분에 걸리는 시간을 10회 측정하여 평균한 것으로 나타내었다. Jacobs 버전 및 제안한 알고리즘 모두 코드 최적화를 수행하지 않고 C언어로 sequence mode(순차적인 모드)로 작성하였고, Intel i7 990X (3.46 GHz), 16GB memory PC에서 수행하였다. (표 1)은 평행빔 구조에서 광선이 균일한 간격으로 512 × 512 크기의 CT 어레이로 입사하는 경우 각 광선에 대한 선적분 시간을 나타내었다. 대략 10%의 속도 향상을 보였다.

(표 1) 512×512 CT 어레이에 대한 평균 선적분 시간

Rays	Views	Jacobs (ms)	New (ms)	Speed up (%)
512	360	160.4±5.57	146.6±4.01	9.4
	720	317.9±5.01	291.8±4.60	8.9
	1440	637.9±5.36	582.9±6.49	9.4
1024	360	318.7±4.88	288.7±3.55	10.4
	720*	634.8±3.94	575.5±4.61	10.3
	1440	1283.0±7.66	1158.2±9.15	10.7

* Rays는 View당 광선의 개수를 의미한다.

4. 토의

Siddon에 의해 제안된 최초 알고리즘에서는 CT 어레이와 광선이 만나는 픽셀의 위치를 계산하는데 많은 시간이 소비되었다. 그 다음으로는 직선 매개변수를 통합하고 오름차순으로 정렬하는 부분이다. Siddon 방법을 개선한 Jocobs 버전은 병목으로 지목된 부분을 광선의 길이와 픽셀 위치를 초기 픽셀 위치 (x_F, y_F)로부터 업데이트하게 함으로써 계산 시간을 획기적으로 단축했다. Jacobs 버전은 광선이 매 픽셀을 지날 때마다 같은 계산을 반복하며 총

$$N = (x_L - x_F + 1) + (y_L - y_F + 1) \qquad (19)$$

번 수행한다. 하지만 본 논문에서 제안한 새로운 알고리즘은 광선이 지나가는 X 또는 Y 방향에 해당하는 하나 또는 2개 픽셀을 동시에 계산함으로써 계산 루프 횟수를 감소시켰다. 이 알고리즘에서 하나의 광선에 대해 계산하는 반복 횟수는 식(18)로 주어지며 최대 N_x 또는 N_y를 넘지 않는다.

본 논문은 평행빔의 경우에 대해 계산하였다. 부채살빔(fan beam)의 경우 (표 1)의 (*)조건에 대해 순수 Siddon 알고리즘과 제안한 알고리즘으로 계산하면 하나의 사이노그램(Sinogram)을 얻는데 각각 3705.1±7.71 ms와 806.5±7.32 ms이 소요된다. Zhang *et. al.*[6]은 표(1)의 (*)의 조건으로 순수 Siddon 방법과 area integral model에 대해 각각 41.53 s와 10.65 s를 얻었다. 계산에 사용된 PC 사양이 다르기 때문에 직접적인 비교는 어렵지만 순수 Siddon 알고리즘을 기준으로 비교하면 Zhang의 Area integral model는 950.1 ms가 예상되어 본

논문에서 제안하는 새로운 알고리즘이 대략 17.8% 더 빠르다.

5. 결론

반복재구성기법을 이용하는 CT에서 빠른 영상재구성을 위해 투영과 역투역을 빠르게 수행할 수 있는 알고리즘이 필요하다. 제안한 알고리즘은 평행빔 구조에서 Jacobs 버전의 Siddon 알고리즘 보다 대략 10%의 속도 향상을 보였다. 제안한 알고리즘은 순차 모드에서 계산하였지만 병렬 처리에 적합한 알고리즘이다. 따라서 CPU의 multi-threads 및 GPU(graphics processing unit)를 활용할 수 있도록 병렬 코드로 확장한다면 매우 높은 속도 향상을 기대할 수 있을 것이다.

Acknowledgement

이 결과물은 2020년도 대구가톨릭대학교 융합연구비 지원에 의한 것임

참고문헌

[1] Y. Nakayama, K. Awai, Y. Funama, M. Hatemura, M. Imuta, T. Nakaura, D. Ryu, S. Morishita, S. Sultana, N. Sato, and Y. Yamashita, “Abdominal CT with low tube voltage: preliminary observations about radiation dose, contrast enhancement, image quality, and noise”, Radiology, 237, 945 – 951, 2005.

[2] J. K. Kim, J. A. Fessler, and Z. Zhang, “Forward-projection architecture for fast iterative image reconstruction in X-ray CT”, IEEE Transactions on Signal Processing, 60, 5508 – 5518, 2012.

[3] H. Gao, “Fast parallel algorithms for the X-ray transform and its adjoint”, Medical Physics, 39, 7110 – 7120, 2012.

[4] R. L. Siddon, “Fast calculation of the exact radiological path for a three-dimensional CT array”, Medical Physics, 12, 252 – 255, 1985.

[5] F. Jacobs, E. Sundermann, B. D. Sutter, M. Christiaens, and I. Lemahieu, “A fast algorithm to calculate the exact radiological path through a pixel or voxel space”, Journal of Computing and Information Technology, 6, 89 – 94, 1998.

[6] S. Zhang, D. Zhang, H. Gong, O. Ghasemalizadeh, G. Wang, G. Cao, "Fast and accurate computation of system matrix for area integral model-based algebraic reconstruction technique", Optical Engineering, 53, 113101, 2014.