

### 7.3.1 代数迭代重建算法

ECT 重建的另一类算法是以代数方程理论为基础的。我们知道 FBP 重建算法是基于连续的 Radon 积分变换定义的连续算法，而实际采集的投影和需要计算的断层重建图像都是以离散的形式进行运算和存储的，因此在实际的 FBP 算法实现过程中需要进行离散化的步骤。而迭代重建算法则是一开始就将图像  $f(x,y)$  划分为离散的多个像素区域，认为每个像素内部的图像值是均匀分布的，并以  $f_i$  表示第  $i$  个像素中的图像值。同样，我们把探测器平面也划分为多个离散的投影单元，并以  $p_j$  表示第  $j$  个投影单元中的投影值。像素  $i$  对投影单元  $j$  的贡献可以写成

$$p_{ij} = c_{ij} f_i \quad \text{公式 7.1}$$

而投影单元  $j$  中的投影值  $p_j$  是所有像素对该投影单元的贡献之和，因此有

$$p_j = \sum_i p_{ij} = \sum_i c_{ij} f_i \quad \text{公式 7.2}$$

把各个投影单元的方程联立起来，则有：

$$\begin{cases} c_{11}f_1 + c_{21}f_2 + \dots + c_{N1}f_N = p_1 \\ c_{12}f_1 + c_{22}f_2 + \dots + c_{N2}f_N = p_2 \\ \vdots \\ c_{1M}f_1 + c_{2M}f_2 + \dots + c_{NM}f_N = p_M \end{cases} \quad \text{公式 7.3 写成}$$

矩阵向量乘积的形式为

$$\begin{bmatrix} c_{11} & c_{21} & \dots & c_{N1} \\ c_{12} & c_{22} & \dots & c_{N2} \\ \dots & \dots & \dots & \dots \\ c_{1M} & c_{2M} & \dots & c_{NM} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} \quad \text{或 } \mathbf{Cf} = \mathbf{p}, \quad \text{公式 7.4 其中}$$

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{21} & \dots & c_{N1} \\ c_{12} & c_{22} & \dots & c_{N2} \\ \dots & \dots & \dots & \dots \\ c_{1M} & c_{2M} & \dots & c_{NM} \end{bmatrix}, \quad \text{称为系统传输矩阵(System transition Matrix), } \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \quad \text{为待}$$

求的图像向量， $\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}$  为已知的投影向量。这样我们就把图像重建问题转化成了由

公式 7.3 从已知的投影数据  $\{p_j\}$  求未知的图像数据  $\{f_i\}$  的问题。

在公式 7.3 中， $\{c_{ij}\}$  是第  $i$  个像素中的图像值对第  $j$  个投影单元中的投影值的贡献因子，或者说从第  $i$  个像素区域发射出来的光子被第  $j$  个投影单元区域接收到的概率，它是由图像区域的像素划分方式和成像系统的性质决定的。如果认为准直器的响应特性是理想的（即只有平行于准直孔轴向的光子才能通过准直器而被探测器接收到），那么我们可以从投影单

元  $j$  出发作一条投影带，这一投影带与第  $i$  个像素区域的交叠部分的面积与第  $i$  个像素区域的总面积之比就是  $c_{ij}$ ，如图 7.1 所示。

理论上讲，在得到了公式 7.3 后，我们就可以采用解代数方程组的方法来获得断层图像。但是实际上这种做法无法实现。首先，方程组只有当投影数据数目与未知图像像素数目相等时才有唯一解，而实际上并不满足这一条件。其次，核医学成像获得的投影数据  $\mathbf{P}$  中通常存在比较严重的统计噪声，这样当系统传输矩阵  $\mathbf{C}$  的病态性质比较严重时，直接求解的方法会受到噪声的严重干扰，得到的解可能和真实情况相比有很大偏差。因此通常采用迭代的方法进行求解。

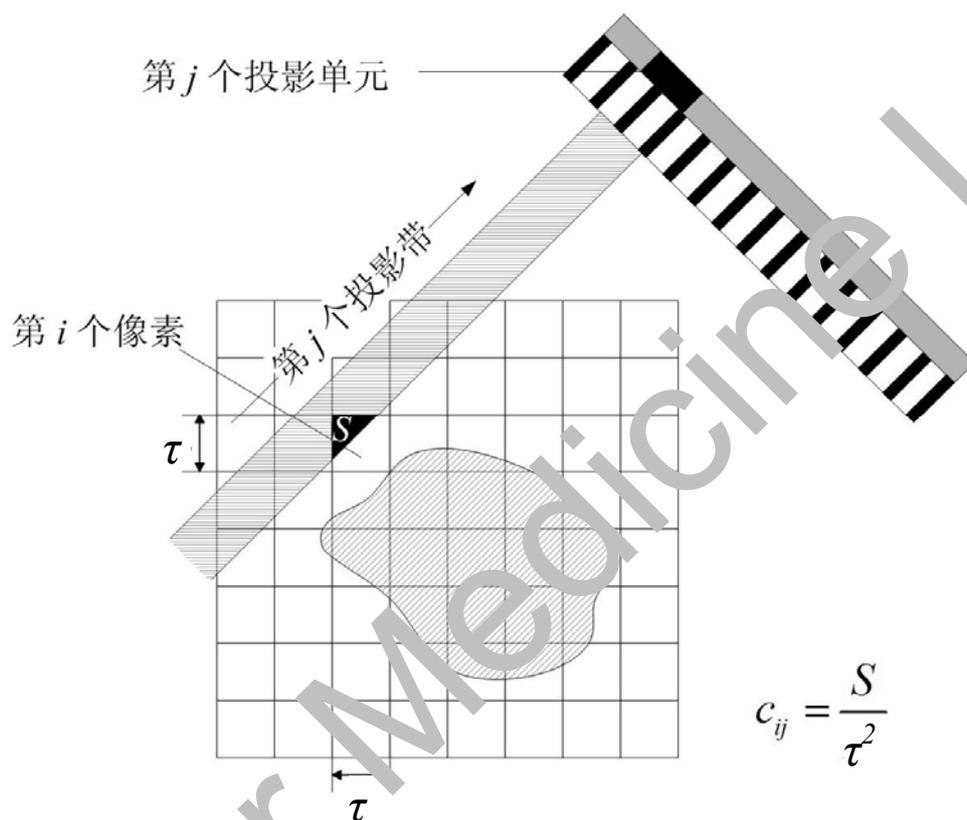


图 7.1: 像素对投影单元贡献因子的计算模型

迭代算法是从断层图像的初始估计值出发，通过对图像的估计值进行迭代修正使其逐渐逼近断层图像的真实值的过程。一般来说，迭代算法的基本步骤如下：首先给待求的断层图像赋予一个初始估计值  $f^0$ ，根据此初始值计算出理论投影值  $p^0$ ，将它和实测投影值  $p$  进行比较，根据一定的原则对初始图像进行修正。然后再根据修正过的图像估计值  $f^1$  计算理论投影值  $p^1$ ，与实测投影值  $p$  比较，再次修正断层图像估计值……如此反复循环，直到相邻两次估计值之差足够小为止。其过程如图 7.2 所示。

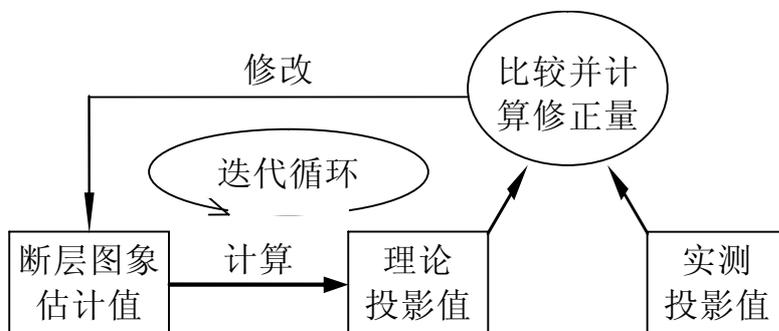


图 7.2: 迭代算法计算过程示意图

和滤波反投影算法相比, 迭代重建算法的优点在于比较灵活, 在投影数据不完备或不均匀分布的条件下也可以实现重建。另外在迭代重建的过程当中可以加入各种约束条件和先验知识 (如图像的边界位置、像素值非负等), 所以重建精度高, 图像质量好。目前人们已经提出了多种迭代算法, 如基于代数方程理论的代数重建技术 (algebraic reconstruction technique, ART) 等算法, 基于统计理论的加权最小二乘算法 (weighted least squares, WLS)、最速下降 (steepest descent) 与共轭梯度 (conjugate gradient) 算法、极大似然一期望最大化算法 (maximum likelihood - expectation maximization, ML-EM) 等。迭代算法之间的区别在于:

- 由断层图像估计值计算理论投影值的方法不同。
- 在修正图像估计值时, 对图像上各像素的修正次序不同。
- 对理论投影值和实测投影值之间的误差进行计算和比较的方法不同。
- 对误差进行修正的实现过程不同。

下面我们将介绍几种常用的代数迭代重建算法。

## 1. ART 算法

ART 算法是一类代数迭代重建算法的统称。ART 算法在迭代修正的过程中, 每轮迭代只依次考虑一个投影单元的投影值, 相应地, 在这轮迭代中只修正对该投影单元有贡献的那些像素的估计值。

最常用的一种 ART 算法是基于 Kaczmarz 提出的交替投影法 (method of projections) 进行迭代修正的。为了说明这种方法的原理, 我们把公式 7.3 式写成:

$$\begin{cases} \mathbf{c}_1 \bullet \mathbf{f} = p_1 \\ \mathbf{c}_2 \bullet \mathbf{f} = p_2 \\ \vdots \\ \mathbf{c}_M \bullet \mathbf{f} = p_M \end{cases}$$

公式 7.5 式

中,  $\mathbf{c}_j = (c_{1j}, c_{2j}, \dots, c_{Nj})$ , “ $\bullet$ ” 是向量点积运算。如果我们以图像  $\mathbf{f}$  的  $N$  个分量构成  $N$  维空间, 那么图像的每一组估计值就是这  $N$  维空间中的一个点, 而线性方程组中的每一个方程  $\mathbf{c}_j \bullet \mathbf{f} = p_j$  都是这  $N$  维空间中的一个  $N-1$  维超平面, 这个超平面与向量  $\mathbf{c}_j$  正交。如果线性方程组有唯一解, 那么这  $N$  个超平面就有唯一的交点, 即我们要求的解。

Kaczmarz 交替投影法的求解思想如下：从初始图像估计值  $f^0$  出发，将这一点投影到第一个方程代表的超平面上，以投影点作为第一次图像估计值  $f^1$ ，把投影点再投影到第二个超平面上获得新的投影点  $f^2$ ，依次类推，直到所有方程的投影都进行完后再返回第一个方程，继续新一轮交替投影。可以证明，如果方程组存在唯一解，那么迭代过程将最终收敛到该点上。

下面我们证明，由  $f^{k-1}$  向第  $k$  个方程投影，获得  $f^k$  的过程可以表示为

$$f^k = f^{k-1} - \frac{(f^{k-1} \cdot c_k - p_k)}{c_k \cdot c_k} c_k \quad \text{公式 7.6}$$

如图 7.3 所示，分别用  $A$ 、 $B$  和  $C$  表示向量  $f^k$ 、 $f^{k-1}$  和  $c_k$  的端点。根据向量的运算法则，有

$$OA = OB - AB \quad \text{公式 7.7}$$

由于  $AB$  和  $OC$  都与超平面正交，故

$$AB = ED = OD - OE \quad \text{公式 7.8}$$

$OD$  和  $OE$  分别是  $f^{k-1}$  和  $f^k$  在  $c_k$  上的投影向量，因此有

$$OD = \frac{f^{k-1} \cdot c_k}{c_k \cdot c_k} c_k \quad \text{公式 7.9}$$

$$OE = \frac{f^k \cdot c_k}{c_k \cdot c_k} c_k \quad \text{公式 7.10}$$

而  $E$  是在超平面  $c_k \cdot f = p_k$  上的点，故

$$OE = \frac{p_k}{c_k \cdot c_k} c_k \quad \text{公式 7.11}$$

将公式 7.9 和公式 7.11 代入公式 7.8，再由公式 7.7 可得

$$OA = OB - \left( \frac{f^{k-1} \cdot c_k}{c_k \cdot c_k} c_k - \frac{p_k}{c_k \cdot c_k} c_k \right), \quad \text{即公式 7.6。或者写成}$$

$$f_i^k = f_i^{k-1} - c_{ik} \frac{\sum_{i'} (f_{i'}^{k-1} c_{i'k}) - p_k}{\sum_{i'} c_{i'k}^2} \quad \text{公式 7.12}$$

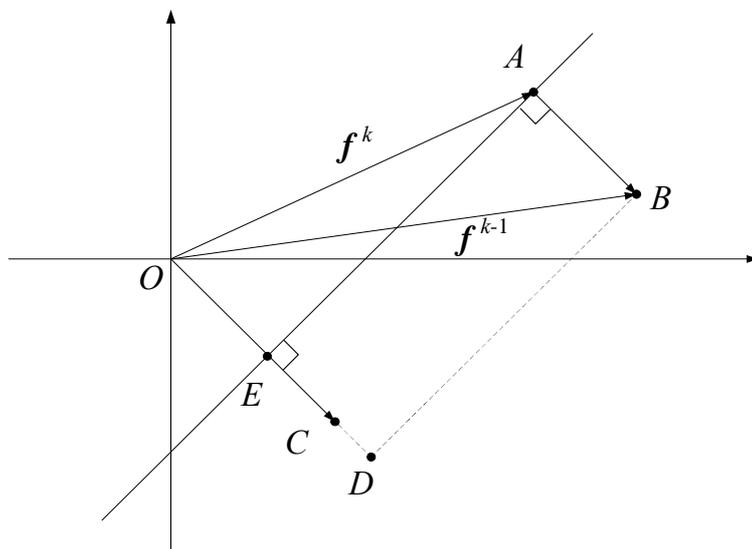


图 7.3: Kaczmarz 算法示意图

下面我们来分析一下公式 7.6 的物理意义。式中， $f^{k-1} \cdot c_k$  是投影步骤，将得到的投影估计值与实测投影值  $p_k$  进行比较，获得投影的修正值  $\Delta p_k = f^{k-1} \cdot c_k - p_k$ 。将投影修正值进行  $1/(c_k \cdot c_k)$  加权，再反投影到对  $p_k$  有贡献的各像素上，对其像素值进行修正。我们注

意到，经过  $1/(c_k \cdot c_k)$  加权因子的作用，在利用像素修正值  $\Delta f_1^k = \frac{\Delta p_k}{c_k \cdot c_k} c_{k1}$ ，

$\Delta f_2^k = \frac{\Delta p_k}{c_k \cdot c_k} c_{k2}$ ，... 求投影时，其值正好等于  $\Delta p_k$ 。

在实现 ART 算法时，一种最简单的计算  $c_{ij}$  的方法是认为像素中的计数都集中在像素中心上，如果投影带经过该像素的中心，则  $c_{ij} = 0$ ，否则  $c_{ij} = 1$ 。这样 ART 算法的公式可以写作

$$\Delta f_k = \frac{p_k - q_k}{N_k} \quad \text{公式 7.13}$$

式中， $N_k$  是  $p_k$  的投影带经过的像素中心个数， $q_k$  是根据第  $k-1$  次图像估计值计算的投影估计值

$$q_k = f^{k-1} \cdot c_i = \sum_{i'=1}^N f_{i'}^{k-1} c_{i'k}。$$

图 7.4 所示为 ART 迭代重建算法的一个简单示例。

ART 算法重建的结果经常会受到椒盐噪声(salt and pepper noise)的影响。产生这一噪声的原因之一是公式 7.13 中对  $c_{ij}$  的近似过于粗略，使重建图像上产生伪像。如果把公式 7.13 改为

$$\Delta f_k = \frac{p_k}{L_k} - \frac{q_k}{N_k}, \quad \text{公式 7.14}$$

则可以在一定程度上减轻伪像，其中  $L_i$  是射线带中心在像素区域中经过的长度被像素宽度  $L$  归一化后的长度。

通过在迭代过程中加入松弛因子  $\omega$  的方法可以减轻噪声的影响。这时公式 7.6 改写为

$$f^k = f^{k-1} - \omega \frac{(f^{k-1} \cdot c_k - p_k)}{c_k \cdot c_k} c_k \quad \text{公式 7.15}$$

其中  $\omega$  是一个 0 到 2 之间的正数。有时还在迭代过程中逐渐改变  $\omega$  的值，可使重建图像质量进一步提高，但这样做的代价是收敛速度变慢。

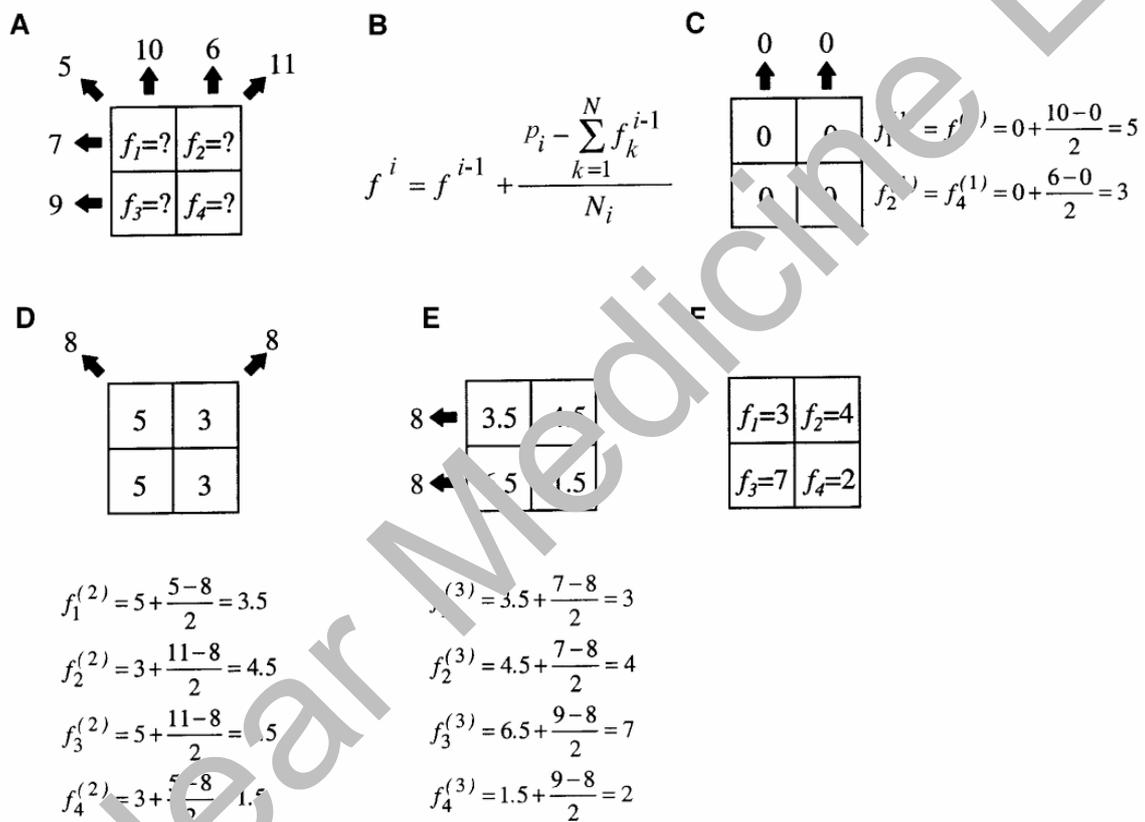


图 7.4: ART 迭代重建算法示例

## 2. SART 算法

在 ART 算法中，每次对图像值进行修正时只依赖于一条投影带上的数据。在其基础上人们提出了同时代数重建算法(simultaneous algebraic reconstruction technique, SART), 其基本思想是在修正某个像素单元的图像值之前，计算该像素单元有贡献的所有投影单元上的投影估计值和实测投影值之间的差别并求其平均值，再利用该平均值对图像值进行修正。SART 算法的迭代公式如下：

$$f_i^k = f_i^{k-1} - \omega \frac{\sum_j \frac{c_{ij}}{\sum_{i'} c_{i'j}} \left[ \left( \sum_{i'} f_{i'}^{k-1} c_{i'k} \right) - p_k \right]}{\sum_j c_{ij}}$$

公式 7.16

$$= f_i^{k-1} - \omega \frac{\sum_j \frac{c_{ij}}{\sum_{i'} c_{i'j}} \Delta p_k}{\sum_j c_{ij}}$$

这样通过把各条投影带上的贡献平均化的方法,可以避免一条投影带上的误差对重建结果带来过大的影响,从而抑制重建图像中的噪声。

Nuclear Medicine Lab.