Advances in Applied Mathematics 应用数学进展, 2017, 6(4), 496-503
Published Online July 2017 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2017.64059

Hans汉斯

Formulas of the Length and Index for the Intersection of a Line and a Fixed Grid in Three-Dimensional Space

Xiaoyuan Zhang, Caifang Wang, Xudao Yin, Zhaoliang Xu

College of Arts and Sciences, Shanghai Maritime University, Shanghai
Email: cfwang@shmtu.edu.cn

Received: Jun. 23rd, 2017; accepted: Jul. 15th, 2017; published: Jul. 18th, 2017

Abstract

In this paper, we introduce the principle of imaging for X-ray CT and try to convert the reconstruction problem to a linear system *Ax=b*. We emphasize on the construction of *A*. Using a reconstruction example, we provide formulas of length and index of the intersection of a line and a fixed grid in three-dimensional space. Finally, we try to verify the formulas with MATLAB programming.

Keywords

Digital Image Reconstruction, X-Ray CT, Grid in 3D, Index

直线与三维固定网格的交线长度与索引公式计算

张晓媛，王彩芳，殷绪导，徐兆亮

上海海事大学文理学院，上海
Email: cfwang@shmtu.edu.cn

收稿日期：2017年6月23日；录用日期：2017年7月15日；发布日期：2017年7月18日

摘　要

本文首先介绍了数字图像重建中最典型的计算机断层成像技术(X-ray Computerized Tomography)的成像原理，并考虑将重建问题转化成线性方程组$Ax = b$的求解，重点讨论了系数矩阵A的生成。以三维实

文章引用: 张晓媛, 王彩芳, 殷绪导, 徐兆亮. 直线与三维固定网格的交线长度与索引公式计算[J]. 应用数学进展, 2017, 6(4): 496-503. https://doi.org/10.12677/aam.2017.64059

际重建问题为例，本文给出了计算直线与三维固定网格的交线长度与索引公式。最后利用MATLAB实现这一计算过程，并分析算法运行的效率。

关键词

数字图像重建，X-Ray CT，三维固定网格，索引

1. 引言

数字图像重建是指根据物体边界测量数据，获取物体内部结构的一种方法[1]。常见的成像技术有全息成像，电子显微，数码相机，核磁共振，超声波，计算机断层层析成像技术等等。图像重建最典型的应用是医学上的计算机断层层析成像技术[2]，这是一种电子计算机技术与X射线检查技术相结合的产物，因其扫描时间快，图像清晰等特点，在成像领域有着非常重要的作用。

X 射线在物体内部沿直线 L 传播，衰减满足 Beer 定理[3]

$$I = I_0 * \exp\left\{-\int_L \mu(x,y,z)\,\mathrm{d}l\right\},$$

其中 $\mu(x,y,z)$ 为物体对 X 射线的衰减系数，I_0 为入射光强度，I 为出射光强度。X-Ray CT 利用这一原理，获取多组入射光强 $I_{0,i}$ 和出射光强度 I_i，得到一系列的投影数据 $b_i = \ln\left(I_{0,i}/I_i\right)$，并根据投影数据值重建物体内部的吸收系数 $\mu(x,y,z)$.

常见的 X-Ray CT 重建算法有迭代算法[4]，分析算法(滤波反投影[5]，反投影滤波[6])等。随着计算机技术的高速发展，迭代重建算法因其利用计算机运算速度快，适合做重复性操作的特点被人们更广泛地应用在图像重建中。迭代算法首先涉及到问题的离散化，以三维图像为例，当待测物体所在的区域被离散成三维固定网格后(共 J 个像素)，如图 1 所示，假设物体在离散后每个格点上的吸收系数是一个常数 x_j，沿着第 i 条射线上的投影值 b_i，可表示成 $b_i = \sum_{j=1}^{J} a_{ij} x_j$，其中 a_{ij} 为第 i 条射线与第 j 个像素的交线长度，这样就将重建问题转化成 $Ax = b$ 的求解[7] [8]。我们的任务是确定矩阵 A 的第 i 个行向量中的非零元以及非零元所在列的索引。

作者吴大瑞，何钦铭在[9]中给出了计算直线与二维固定网格交线长度的算法，我们将这一算法进行推广，给出了计算直线与三维固定网格的交线长度的算法。此外，我们也给出了直线与三维固定网格的交线索引的公式。最后我们通过 MATLAB 实现这一算法[10] [11]，并用两组数值实验分别验证了算法的正确性和有效性。

2. 计算直线与网格的交线长度及索引位置

2.1. 空间坐标系的建立及图像的离散化

为后续描述方便，我们首先定义物体的参数(见表 1)。

1) 假设待测物体处于一个长、宽、高分别为 l_x，l_y，l_z 的矩形区域内。以矩形的中心为原点建立空间直角坐标系，对矩形区域进行离散化后得每个格点的长、宽、高分别为 $xUnit = l_x/xRes$，$yUnit = l_y/yRes$，$zUnit = l_z/zRes$。

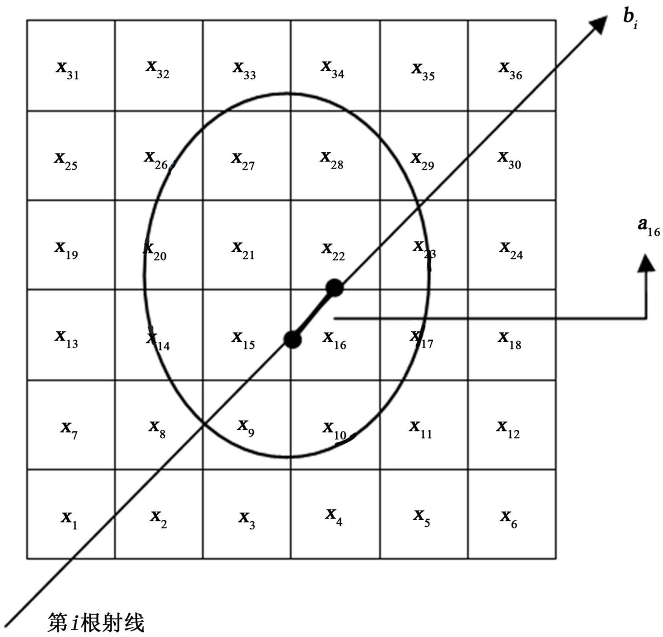

Figure 1. Discretization of the object to be imaged
图 1. 待测物体离散化

Table 1. Parameters of the object to be imaged
表 1. 待测物体参数

变量	意义
l_x	待测物体的长
l_y	待测物体的宽
l_z	待测物体的高
$xRes$	待测物体 x 方向上的分辨率，文中为偶数
$yRes$	待测物体 y 方向上的分辨率，文中为偶数
$zRes$	待测物体 z 方向上的分辨率，文中为偶数

2) 我们将网格中格点依 z 轴依次从下往上逐层标号，以最底层格点为例对网格进行标号，从数字 1 开始，按图 2 的方法，直到标完对应的格点数 $xRes * yRes$；按同样的方法在矩形区域内向上一层对格点进行标号，依次向上标，直到标完对应的格点数 $xRes * yRes * zRes$，以此完成对矩形区域的离散化。

2.2. 第 i 根射线所在位置描述

假设第 i 根射线是由第 i 个光源 $(xSource, ySource, zSource)$ 及探测器 $(xDetector, yDetector, zDetector)$ 所确定，则该直线所在的空间直线参数方程可表示为(t 为参数)

$$\begin{cases} x = xSource + mt, \\ y = ySource + nt, \\ z = zSource + pt. \end{cases}$$

(yRes−1)*xRes+1	(yRes−1)*xRes+2	……	xRes*yRes
……	……	……	……
xRes+1	xRes+2	……	2*xRes
1	2	……	xRes

Figure 2. Label of grid on *xoy* plane
图 2. *xoy* 平面网格标号

其中(m,n,p)为直线的方向向量，可以根据确定的光源和探测器位置的坐标确定 $m = xDetector - xSource$，$n = yDetector - ySource$， $p = zDetector - zSource$。为后续描述方便，需保证 m,n,p 按序大于 0，若出现以下情况之一：(1) $m<0$，(2) $m=0$，$n<0$，(3) $m=0$，$n=0$，$p<0$，则令 $(m,n,p)=-(m,n,p)$ 即可。

2.3. 直线与网格交线的长度与索引公式

1) 为确定直线与网格的交线长度，我们首先给出直线与网格的交点。设光源探测器直线与 $x=-l_x/2+xUnit*(k-1)$ 平面的交点分别为 (x_k^1,y_k^1,z_k^1) ($k=1,2,\cdots,xRes+1$)。根据条件

$$\begin{cases} y_k^1 > yRes/2 * yUnit, \\ y_k^1 < -yRes/2 * yUnit, \end{cases} \quad \begin{cases} z_k^1 > zRes/2 * zUnit, \\ z_k^1 < -zRes/2 * zUnit. \end{cases}$$

删除越界的点，得到直线与 x 轴网格线相交 K_1 组的坐标，仍然记为 (x_k^1,y_k^1,z_k^1)。同理，可获得直线与 y 轴网格线相交的 K_2 组有效坐标值，为 (x_k^2,y_k^2,z_k^2)，与 z 轴网格线相交的 K_3 组有效坐标值，为 (x_k^3,y_k^3,z_k^3)。

将三组数据按照 x 坐标从小到大进行重排，若 $m=0$，则按 y 从小到大重排，若 $m=0$ 且 $n=0$，则按 z 从小到大重排。得到一系列不重复交点，记为 (x_t,y_t,z_t) $(t=1,2,\cdots,K)$。则该直线与网格的交线长度可表示为

$$a_{ij}=\sqrt{(x_{t+1}-x_t)^2+(y_{t+1}-y_t)^2+(z_{t+1}-z_t)^2}. \quad (1)$$

2) 为计算直线与网格交线的索引，可以根据交线起点 (x_t,y_t,z_t) 找到该格点中 x,y,z 坐标最小的顶点，并由此顶点推断索引值 j。具体的实现中，首先令

$$P_x=\text{floor}\left(\frac{x_t}{xUnit}\right),\quad P_y=\text{floor}\left(\frac{y_t}{yUnit}\right),\quad P_z=\text{floor}\left(\frac{z_t}{zUnit}\right).$$

- 若 $n<0$，且 $P_y * yUnit = y_t$，则修正 $P_y = P_y - 1$；
- 若 $p<0$，且 $P_z * zUnit = z_t$，则修正 $P_z = P_z - 1$.

图 3 为几种特殊的相交情况，以此说明上述修正。

最后令

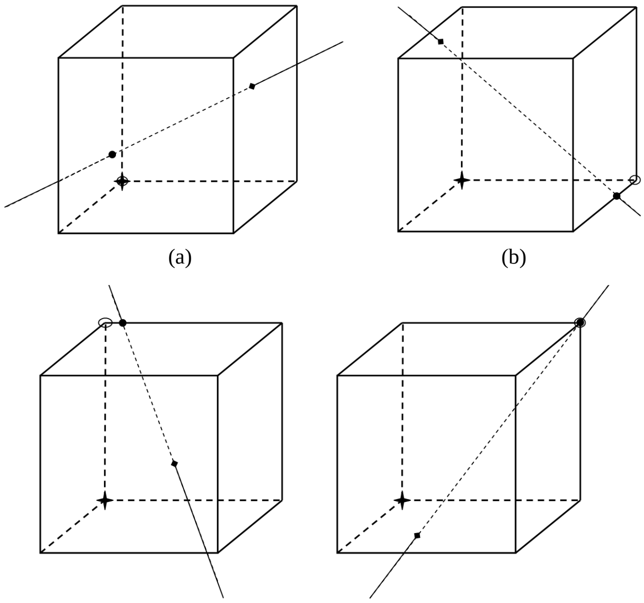

Figure 3. Special cases of intersection of a line and fixed grid. Solid circle: starting point (x_t, y_t, z_t); Rhombus: end point $(x_{t+1}, y_{t+1}, z_{t+1})$; Hollow circle: point before correction $(P_x * xUnit, P_y * yUnit, P_z * zUnit)$; Star: corrected point $(P_x * xUnit, P_y * yUnit, P_z * zUnit)$. (a) $m > 0, n > 0, p > 0$; (b) $m > 0, n < 0$ and $P_y * yUnit = y_t$; (c) $m > 0, n > 0, p < 0$ and $P_z * zUnit = z_t$; (d) $m > 0, n < 0$ and $P_y * yUnit = y_t$, $p < 0$ and $P_z * zUnit = z_t$

图 3. **几种特殊的直线与网格相交情况。实心圆：起点**(x_t, y_t, z_t)；**菱形：终点**$(x_{t+1}, y_{t+1}, z_{t+1})$；**空心圆：修正前的点**$(P_x * xUnit, P_y * yUnit, P_z * zUnit)$；**星形：修正后的点**$(P_x * xUnit, P_y * yUnit, P_z * zUnit)$。(a) $m > 0, n > 0, p > 0$；(b) $m > 0, n < 0$ 且 $P_y * yUnit = y_t$；(c) $m > 0, n > 0, p < 0$ 且 $P_z * zUnit = z_t$；(d) $m > 0, n < 0$ 且 $P_y * yUnit = y_t$， $p < 0$ 且 $P_z * zUnit = z_t$

$$
\begin{aligned}
P_x &= P_x + xRes/2,\\
P_y &= P_y + yRes/2,\\
P_z &= P_z + zRes/2.
\end{aligned}
$$

根据修正后的(P_x, P_y, P_z)，可得交线所在格点的索引公式为

$$j = (P_x + 1) + P_y * xRes + P_z * xRes * yRes. \tag{2}$$

3. 数值实验

为验证算法的有效性和正确性，我们进行如下两组数值实验。算法在一台 Dell 电脑上完成，操作系统为 Windows 7 旗舰版(64 位)，处理器：Intel(R) Core(TM) i7-4790，CPU@3.60 GHz，RAM:8 GB。采用 Matlab R_2013a 版本。

3.1. 算法正确性验证

设 $xRes = 4$， $yRes = 4$， $zRes = 4$， $l_x = 4$， $l_y = 4$， $l_z = 4$。分别取如下 4 组光源和探测器。经程序计算得到结果如表 2 所示。此外，我们对不同光源和探测器情况作了手动计算，经过比较，我们确定两者一致，算法的正确性得到验证。

Table 2. Correctness verification for the numerical method
表 2. 程序正确性检测结果

编号	光源与探测器	a_{ij} 与 *index*
(1)	$(6,4,1)$	$(a_{ij})=[0.6481,0.9721,0.3240,1.2961,1.2961]$
	$(-4,-4,-1)$	$index=[17,18,22,23,44]$
(2)	$(6,-4,1)$	$(a_{ij})=[0.6481,0.9721,0.3240,1.2961,1.2961]$
	$(-4,4,-1)$	$index=[29,30,26,27,40]$
(3)	$(6,4,-1)$	$(a_{ij})=[0.6481,0.9721,0.3240,1.2961,1.2961]$
	$(-4,-4,1)$	$index=[33,34,38,39,28]$
(4)	$(6,-4,-1)$	$(a_{ij})=[0.6481,0.9721,0.3240,1.2961,1.2961]$
	$(-4,4,1)$	$index=[45,46,42,43,24]$

3.2. 算法运行效率

我们模拟现实的光源和探测器分布情况，假设待测物体参数设置为

$$l_x=l_y=l_z=20\text{ cm}$$
$$xRes=yRes=zRes=256.$$

光源和探测平面同时绕待测物体螺旋上升。具体参数见表 3。

初始状态下光源位置设为 $\left(x^{(1)},y^{(1)},z^{(1)}\right)=\left(d,0,Z_0\right)$。与之对应的探测器平面如图 4 所示。

光源和探测器平面同时绕 z 轴旋转，第 k 个光源转过角度为

$$\theta=\alpha(k-1),\quad k=1,2,\cdots.$$

光源位置为 $\left(x^{(k)},y^{(k)},z^{(k)}\right)$

$$\begin{bmatrix}x^{(k)}\\y^{(k)}\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x^{(1)}\\y^{(1)}\end{bmatrix},$$
$$z^{(k)}=z^{(1)}+\frac{\theta}{2\pi}h.$$

与之相对应的探测器上各点坐标为 $\left(X_{ij}^{(k)},Y_{ij}^{(k)},Z_{ij}^{(k)}\right)$

$$\begin{bmatrix}X_{ij}^{(k)}\\Y_{ij}^{(k)}\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}X_{ij}^{(1)}\\Y_{ij}^{(1)}\end{bmatrix},$$
$$Z_{ij}^{(k)}=Z_{ij}^{(1)}+\frac{\theta}{2\pi}h.$$

由于系数矩阵是一个大型稀疏矩阵，在实际的迭代过程中，如果将 A 存入内存，是一个耗空间的做法；如果每次迭代都从外设读取数据，也是一个耗时的做法。因此我们考虑在每步迭代中，都计算一次系数矩阵。模拟实际情况，我们假设共采集 2.7×10^5 次数据，形成 2.7×10^5 组交线长度和索引标号。

……	……	……	……
$\text{Detector}_{1j}^{(l)}$	……	$\text{Detector}_{ij}^{(l)}$	……
……	……	……	……
$\text{Detector}_{11}^{(l)}$	……	$\text{Detector}_{i1}^{(l)}$	……

Figure 4. Label of detector
图 4. 平面探测器编号

Table 3. Parameters for sources and detectors
表 3. 光源、探测器参数

变量	意义	数据
d	光源到待测物体中心轴的距离	60 cm
D	探测平面中心到待测物体中心轴的距离	40 cm
Lx	平板探测器长度	40 cm
Ly	平板探测器高度	40 cm
$XRes$	平板探测器水平方向分辨率(偶数)	50
$YRes$	平板探测器竖直方向分辨率(偶数)	50
α	采集数据的旋转角(弧度)	$\pi/18$
h	螺距	10 cm

采用上述程序，在 MATLAB 下运行，实际代码执行的时间为 597.7133 秒。由此，我们也可以大体估计每次迭代所用时间。根据数值实验的结果，我们发现运行效率可以接受。

4. 总结

本文以 X-ray CT 为例，研究重建问题中系数矩阵的生成方式，将系数矩阵元素和直线与三维固定网格相交联系起来，研究了直线与三维固定网格的交线长度与索引。最后用 MATLAB 程序语言实现了这一算法，并将算法应用到三维重建的系数矩阵生成上，完成一次矩阵的计算时间在 10 分钟以内。运算效率是可行的。

基金项目

国家自然科学基金青年基金项目(11401372)。

参考文献 (References)

[1] 冈萨雷斯. 数字图像处理[M]. 北京: 北京电子工业出版社, 2009.
[2] 曾晖, 孙腊珍, 汪晓莲. CT 计算机断层扫描成像实验[J]. 物理实验, 2008, 28(12): 9-12.
[3] 庄天戈. CT 原理与算法[M]. 上海: 上海交通大学出版社, 1992.
[4] 潘晋孝. X 射线 CT 迭代算法研究[M]. 北京: 北京大学数学科学学院, 2006.
[5] 张顺利, 李卫斌, 唐高峰. 滤波反投影图像重建算法研究[J]. 咸阳师范学院学报, 2008, 23(4): 47-49.
[6] 洪贤勇, 乔志伟. 用反投影滤波算法实现 CT 图像的 ROI 重建[J]. 电视技术, 2014, 38(7): 29-32.

[7] 陈洪磊，贺建峰，刘俊卿，马磊. 迭代图像重建中系统矩阵与重建图像质量关系研究[J]. 计算机应用，2013, 33(1): 53-56, 68.

[8] 王亮，寿永熙，秦俊平. 图像重建迭代算法的研究[J]. 黑龙江科技信息, 2007(21): 72-72.

[9] 吴大瑞，何钦铭. 一种简单的基于固定网格的空间直线索引算法[J]. 江南大学学报(自然科学版)，2005, 4(4): 394-396.

[10] 张振东，哈力旦・A. 基于 MATLAB 的 CT 的图像三维重建的研究与实现[J]. 电子世界, 2013(3): 87-88.

[11] 曾筝，董芳华，陈晓，周宏，周建中. 利用 MATLAB 实现 CT 断层图像的三维重建[J]. CT 理论与应用研究, 2004, 13(2): 24-29.