

赛区评阅编号（由赛区组委会填写）：

---

## 2017 年高教社杯全国大学生数学建模竞赛

### 承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称“竞赛章程和参赛规则”，可从 <http://www.mcm.edu.cn> 下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、“贴吧”、QQ 群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：\_\_\_\_\_ A \_\_\_\_\_

我们的报名参赛队号（12 位数字全国统一编号）：\_\_\_\_\_ 111122223333 \_\_\_\_\_

参赛学校（完整的学校全称，不含院系名）：\_\_\_\_\_ 西安交通大学 \_\_\_\_\_

参赛队员（打印并签名）：1. \_\_\_\_\_ 柏思特 \_\_\_\_\_

2. \_\_\_\_\_ 王则骁 \_\_\_\_\_

3. \_\_\_\_\_ 职 卓 \_\_\_\_\_

指导教师或指导教师组负责人（打印并签名）：\_\_\_\_\_ 陈 磊 \_\_\_\_\_

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期：\_\_\_\_\_ 2017 年 9 月 17 日 \_\_\_\_\_

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

# CT 系统参数标定及成像

## 摘要

针对 CT 系统的参数标定及成像问题, 本文首先根据已知模板信息分别确定了旋转中心在正方形托盘中的位置、探测器单元的间距以及该 CT 系统旋转的 180 个方向角度; 其次, 基于上述结果确定了附件 3 中未知介质的位置、几何形状以及吸收率, 并给出了待求 10 个位置的吸收率; 之后, 本文进一步地求解出了附件 5 中未知介质的相关信息; 最后, 分析了 CT 系统的精度与稳定性, 设计了改进后的新模板与标定模型。

对于问题一, 为了标定 CT 系统旋转中心等参数, 首先, 以待测物 X 射线投影的理论值与附件 2 中接收信息的实际值的欧氏距离作为目标函数, 以 CT 系统的旋转方向为约束条件, 建立了关于 CT 系统待标定参数的优化模型。之后, 利用遗传算法, 在改进 MATLAB2017a 自带的 Radon 函数程序后, 基于附件 1、附件 2 中的数据, 以参数的近似计算值为初值, 对优化模型进行求解。最后, 求解得 CT 系统旋转中心的坐标为 (46.6736, 54.5797), 探测器单元间距为 0.2765mm, X 射线的 180 个方向分别为: 以 X 射线入射方向与椭圆物体短轴的夹角表示, 从  $-60.2967^\circ$  到  $118.5965^\circ$  的 180 个非等间隔的角度方向。

对于问题二, 为确定未知介质的几何形状及接收率等信息, 本文建立了反投影模型, 采用了滤波反投影算法, 基于附件 3 的数据, 利用 MATLAB2017a 软件, 求解未知介质信息; 之后, 得到了未知介质的位置、几何形状和吸收率等数据, 并计算出附件 4 中 10 个位置的吸收率分别为: 0.0000, 2.6022, 6.5378, 0.0000, 0.1345, 3.1339, 5.9121, 0.1198, 9.1801, 0.1293。

对于问题三, 本文沿用问题二中的反投影模型, 首先, 利用滤波反投影算法, 基于附件 5 中的数据, 对模型进行求解, 得到了未知介质的几何形状与吸收率分布情况; 之后, 考虑了该未知介质形状结构复杂的特点, 基于前人的研究, 对算法进行了改进, 显著提升了图像的清晰度与吸收率的精确度; 最后, 求解得到附件 4 中 10 个位置的吸收率分别为: 0.0000, 2.6022, 6.5378, 0.0000, 0.1345, 3.1339, 5.9121, 0.1198, 9.1801, 0.1293。

对于问题四, 首先基于问题一的结果计算了各参数的相对误差, 得到了其对应精度。其次, 通过分析多组参数适应度函数值的波动情况, 定性得出了该参数标定方法稳定性较优的结果, 在进一步分析误差和波动产生原因的基础之上, 得出标定方法仍有提升空间的结论; 因此, 设计了由两个匀质圆形构成的改进模板和对应的标定模型; 最后, 说明了改进模板具有稳定性好、误差小、运行效率高的优点。

**关键字:** CT 系统, 参数标定, 优化模型, 遗传算法, 滤波反投影算法, 吸收率

## 一、问题背景与分析

电子计算机断层扫描技术 (Computed Tomography), 即 CT, 兴起于上世纪 70 年代。X 射线计算机断层成像技术 (X-ray CT) 是一种极为重要的诊疗手段和一种新型的无损检测技术, 可以在不损坏样品的前提下, 利用不同样品对 X 射线吸收特性的差异, 对生物组织或工程材料进行断层成像, 进而得到样品的内部结构信息<sup>[1]</sup>。由于 CT 技术独特的成像优势, 其在医学和工业领域正得到愈发广泛的应用, 发挥着极大的作用, 成为医学和工业界诊断和检测的主流技术之一。仪器检测对 CT 系统的精度要求极高。因此, 研究 CT 系统各参数的标定以及成像的机理过程有着重要意义。

在目前已有的研究中, 随着时间的推移与科技的发展, 在提高 CT 系统的精度方面已有许多科研机构与技术企业做出了重要成果: 自 1970 年间第一台临床用 CT 装置由英国 EMI 公司研制成功<sup>[2]</sup>以来, X 射线的发射与扫描方式不断被改进<sup>[3]</sup>。2006 年西门子推出世界上第一台双能 CT 系统, 丰富了断层信息, 加快了扫描速度<sup>[4]</sup>。但是由于 CT 系统运作机理复杂、精度要求高、成像算法耗时长, 仍存在许多待解决的问题。针对 CT 系统旋转中心的确定, 刘明进就工业 CT 旋转中心定位的问题, 结合不同中心偏差情况与成像结果数据, 基于点扩展函数拟合出了中心位置与成像偏差的关系<sup>[5]</sup>, 但对实验案例的数量要求高, 未总结出根据探测器读数确定旋转中心的一般方法; 孟凡勇等提出了一种利用投影原始数据精确确定 CT 投影中心的方法, 通过分析像素点数值的相关性及噪声确定中心偏移情况<sup>[6]</sup>, 但在未知选专家杜的情况下难以实践; 对于探测器单元间距以及旋转角度的确定, 大多情况下采用等距离等角度的设计, 关于未知间距未知方向的情况研究较少; 针对未知介质成像的问题, 毛小渊采用了滤波反投影方法, 研究了扇束投影数据的重排算法<sup>[7]</sup>, 但对平行入射的 X 射线成像方式未做讨论; 郭静钰以 Shepp-Logan 头部模型为例, 比较了不同方法的成像效果<sup>[8]</sup>, 但不同方法各有优劣, 其适用情况在文中未做讨论; 对于 CT 系统标定模板的设计问题, 相关的实例较少。总而言之, 虽然 CT 技术发展日渐成熟, 但对于未知中心、探测器单元间距、旋转角度的 CT 系统标定问题, 尚无既定的解决方案, 对于基于投影数据的成像问题还没有完善的算法方案。

针对 CT 系统参数标定及成像的问题, 本文从其旋转中心位置、探测器单元间距、X 射线旋转方向等相关参数的标定, 未知介质位置信息、几何形状、吸收率的计算以及参数标定的精度分析、优化模板设计和标定模型等方面加以分析和讨论。

问题一, 考虑到本文只需要涉及 CT 系统及待测物体的二维性质, 故可以先构建 CT 系统的平面几何模型, 并将其解析化, 以便问题的分析; 之后, 为标定系统各参数的精确值, 可以将待求参数计算所得的理论接收信息与实际接收信息差值最小作为目标函数, 建立优化模型, 通过遗传算法智能搜索出最优解, 进而得到相关参数的实际精确值。

问题二, 首先考虑到附件三中的投影数据为离散的矩阵形式, 猜想可将待求图像表示为离散的图像向量, 并用建立在离散模型上的迭代重建算法进行求解。考虑到, 迭代重建算法需要建立每条射线和每个像素点之间的关系, 问题规模可能过于庞大, 不便求解。故可基于解析法, 建立反投影模型, 通过离散形式的傅里叶变换与积分对本题进行求解。

问题三, 考虑到附件 2、附件 3、附件 5 的数据形式相同, 对于附件 5 中未知介质的图像信息确定, 可以直接采用问题二中的模型和算法求解问题。之后, 可以分析附件 5 中未知介质的结构特点, 对算法加以改进, 得到更加精确的吸收率分布结果。

问题四, 对于参数标定的精度的分析, 可通过解得的多组参数值, 利用相对误差

定义来反映。对于参数标定的稳定性分析，可利用遗传算法的适应度函数值变化定性分析。在分析了误差和不稳定性来源后，设计更完善的模板及其对应的标定模型。

## 二、符号说明

符号	含义	单位
$I_0$	穿过物质前的射线强度	$W/m^2$
$I$	穿过物质之后的 X 射线强度	$W/m^2$
$P$	CT 系统所接收的投影数据	-
$P_{\theta_m, i}^*$	第 $m$ 个角度下第 $i$ 条射线的投影数据	-
$\mu$	某物质的线性衰减系数	$m^{-1}$
$\alpha$	某物质的吸收率	-
$k$	线性衰减系数与吸收率的比例系数	$m$
$\theta_m$	旋转过程中第 $m$ 个角度	度
$d_n$	探测器单元间距	mm
$d_0$	基准线到探测器左端的距离	mm
$F_{\theta}(\omega)$	$\theta$ 投影角下投影的傅里叶变换	-
$f(x, y)$	FBP 算法中待求的图像函数	-
$F(u, v)$	$f(x, y)$ 的二维傅里叶变换	-
$f_{cal}(x, y)$	重建后的图像函数	-
$R$	改进的标定模板中圆的半径	mm
$l_{00'}$	改进的标定模板中两圆心间距	mm
$l_{a_i, a_j}$	标号为 $i$ 与 $j$ 的探测器间距	mm

## 三、模型假设

- 1) 假设附件中提供的所有数据均真实且准确。
- 2) 假设不考虑 X 光射线发生衍射的情况。
- 3) 假设每一个探测器单元可看作一个点。

## 四、模型的建立与求解

针对 CT 系统的参数标定及成像问题，本文首先对 CT 系统工作机理加以说明，并对问题中的一些关键概念做出阐释，为建立模型做出准备；其次，根据附件 1 中已知的标定模板信息，通过对问题的几何化分析，建立了优化模型，并采用遗传算法进行

求解，分别确定了旋转中心在正方形托盘中的位置、探测器单元的间距以及该 CT 系统旋转的 180 个方向角度；之后，基于上述问题的求解结果，采用图像向量模型对未知介质图像信息建模，并分析了该模型求解的复杂性，换用反投影模型及滤波反投影算法对问题进行求解，确定了附件 3 中未知介质的位置、几何形状以及吸收率，同时给出待求 10 个位置的吸收率；进一步地，基于上述问题所建立的反投影模型，基于附件 5 的数据，求解出了未知介质的相关信息，并结合该未知介质结构的复杂性及前人的研究，对算法做出改进，优化了图像的清晰度与吸收率的准确度；最后，本文通过计算相对误差以及适应度函数值，分析了 CT 系统参数标定的精度与稳定性，设计了新模板与标定模型。

#### 4.1 模型准备

在对问题进行求解之前，有必要对问题所涉及到的 CT 的工作原理和题目中的关键词“吸收率”和“探测器接收信息”进行分析，以方便问题的解决。

##### 4.1.1 CT 系统的工作原理

在 CT 系统参数标定及成像问题中，所讨论的 CT 系统以平行光束的扫描形式对投影信息进行二维重建。系统的成像的原理是根据 X 射线照射不同的物质衰减不同的特点，可通过采集不同角度的投影信息来计算物体的衰减系数分布，并以此反映物体形状。故 CT 重建的图像应是衰减系数 $\mu$ 的分布<sup>[2]</sup>。

##### 4.1.2 “吸收率”和“探测器接受信息”的定义

对于均匀物质，X 射线的变化规律满足指数衰减，即朗伯比尔定律 (Lambert-Beer's Law)<sup>[9]</sup>：

$$I = I_0 e^{-\mu \Delta x} \quad (1)$$

其中， $I_0$ 为穿过物质前的射线强度； $I$ 为穿过物质之后的 X 射线强度， $\mu$ 为该物质的线性衰减系数， $\Delta x$ 为射线穿过的物质厚度。又由参考文献<sup>[7]</sup>可知，CT 系统的投影数据 $P$ 定义如下：

$$P = \ln \left( \frac{I_0}{I} \right) = \int_L \mu(l) dl \quad (2)$$

其中， $P$ 为 CT 系统所接收的投影数据， $\mu(l)$ 为沿 X 射线方向关于穿过物质路径 $l$ 的衰减系数函数。

通过对数据的分析，考虑到题目中 CT 系统投影数据经过增益处理，本文认为题目所定义的吸收率 $\alpha$ 即附件一中数据与衰减系数 $\mu$ 在比例系数 $k$ 的关系，即：

$$\alpha = k \cdot \mu \quad (3)$$

由此可知探测器所探测的数据 $P$ 也与附件二中模板接收并增益处理后的信息 $P_{\theta_m, i}^*$ 存在比例系数关系，即：

$$P_{\theta_m, i}^* = k \cdot P \quad (4)$$

并在下文的问题求解过程中证实了该定义式的合理性，得到了理想的结果。

#### 4.2 问题一：CT 系统旋转中心、探测器距离及旋转角度的标定

针对确定 CT 系统选旋转中心、探测器单元间距及 X 射线旋转角度的问题，可以利用平面几何的分析方法，建立优化模型求解。考虑待测物体的二维性质，可以将 CT 成像过程抽象为平面几何问题。先在 CT 系统示意图中建立平面直角坐标系，并将 X

射线、待测物体等解析化，列出其坐标位置或直线表达式。后对 CT 系统成像工作机理进行分析，计算探测器接收信息的理论值，以该理论值与附件二中的实际值误差最小为目标函数，以 CT 系统工作机理为约束条件，建立优化模型。最后，利用遗传算法，反解旋转中心位置、探测器单元间距以及 X 射线旋转角度等信息。

#### 4.2.1 CT 系统的参数标定的数学问题转化

##### 1) CT 系统的平面几何模型及 X 射线的解析表达

由于在 CT 系统参数标定及成像问题中，待测介质均为二维形式，检测仪器的三维性质也未被利用，因此，可以将 CT 系统以抽象为平面几何的模型，以简化问题的分析。

进一步地，为方便具体问题的分析，将该平面模型解析化。以上图中正方形托盘的左下角为原点，以其底边为 x 轴，建立平面直角坐标系，如下图：

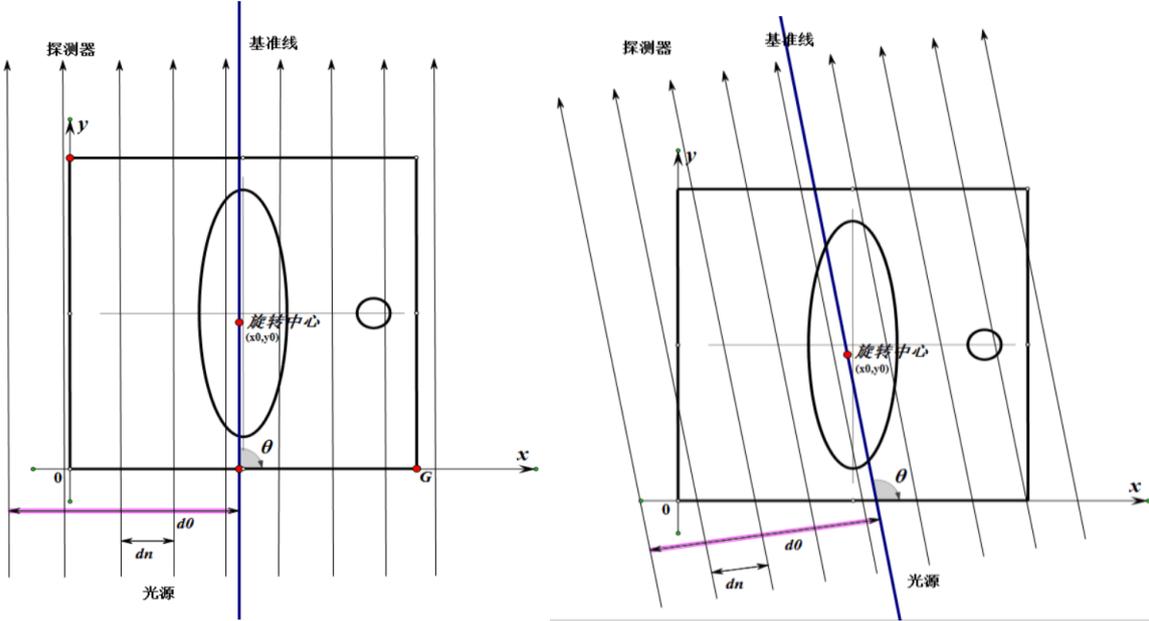


图 4.1 CT 系统各参数的解析化表示图

下面分别对各相关参数进行解析化处理。设 CT 系统旋转中心坐标为 $(x_0, y_0)$ 。记探测器单元间距为 $d_n$ ，记基准线到探测器左端的距离为 $d_0$ 。定义一条基准线 $L_0$ 为过旋转中心且垂直于探测器平面的直线。定义旋转角度 $\theta$ 为基准线与待测椭圆物体短轴的夹角，具体地，在旋转过程中，第 $m$ 个角度为 $\theta_m$ ， $m = 1, 2, \dots, 180$ 。

因此，基准线的方程为：

$$y - \tan\theta \cdot x - x_0 - y_0 = 0 \quad (5)$$

由于所有 X 射线均与探测器垂直，因而与基准线平行，故第 $i$ 条 X 射线 $L_i$ 的方程为：

$$y - \tan\theta \cdot x + \tan\theta \cdot x_0 + \tan\theta \cdot \sin\theta \cdot d + \cos\theta \cdot d - y_0 = 0 \quad (6)$$

其中 $d = (i - 1) \cdot d_n - d_i$ 。

以上即为对 X 射线的解析化表示。

##### 2) 对待测物体信息的解析表达——Radon 变换

由于图像重建所用到的数据都是通过对物体的扫描获得的，本文引入 Radon 变化的相关知识。奥地利数学家 Radon 在 1917 年提出了投影图像重建的基本数学理论，为 CT 技术建立了数学理论基础<sup>[10]</sup>。由 4.1.2 接受信息 $P_{\theta_m, i}^*$ 的定义可知，接收信息

$P_{\theta_m, i}^*$ 实际上是 X 射线衰减系数的线积分。而 Radon 变化在数学上可以表示为已知函数

沿直线路径的线积分，定义式为：

$$\rho_{\theta}(t) = \int_L f(x, y) ds \quad (7)$$

上式为 Radon 变换的定义式，L 为投影射线。Radon 变换的示意图如下：

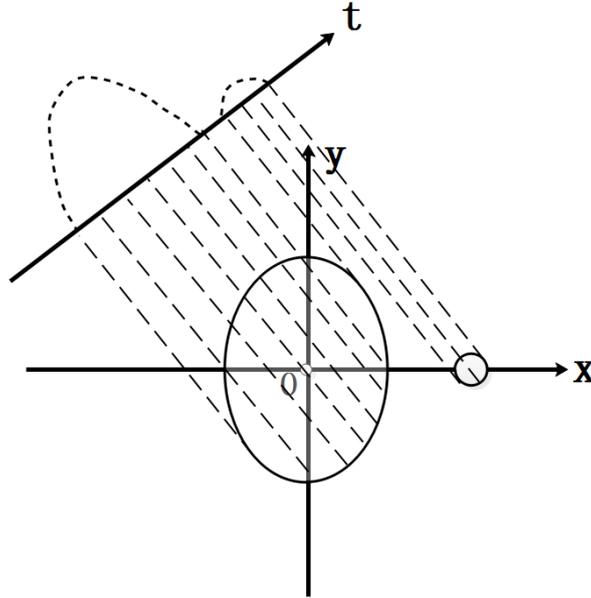


图 4.2 Radon 变换示意图

从数学理论的角度而言，Radon 变换是  $(x, y)$  到  $(t, \theta)$  的空间变换， $t$  为旋转坐标轴的横轴坐标， $\theta$  为旋转角度。在计算机实现中，由于图像的信息以像素点的离散形式给定，只需对射线穿过的素值按长度或面积分配后累加<sup>[8]</sup>。

由于 MATLAB 自带的 Radon 函数无法设定投影射线的解析信息，本文在此基础上进行了改进，即可让改进后的 Radon 函数在给定系统参数的情况下，相应地输出 512 个探测器的接收信息。因而，可以方便地进行任一投影方向接收信息的求解，并可嵌入到下文遗传算法的适应度函数当中以适应问题求解的需要，其算法原理在次不作赘述。

#### 4.2.2 优化模型的建立

##### 1) 目标函数的确立

由上文可知，对于每一个旋转角度和每一条 X 射线，都有一个理论计算出的探测器接收值  $P_{\theta_m, i}^*$ 。则，对于每一个旋转角度  $\theta_m$ ，都有向量  $\mathbf{P}_{\theta_m}^* = (P_{\theta_m, 1}^*, P_{\theta_m, 2}^*, \dots, P_{\theta_m, 512}^*)$  表示 512 个探测器应接收到的理论值。在附件二中，可以知道探测器扫描待测物体所接收到的实际值  $\mathbf{P}_{\theta_m} = (P_{\theta_m, 1}, P_{\theta_m, 2}, \dots, P_{\theta_m, 512})$ 。因此，确立目标函数  $f$ ，使理论值与实际值的差异最小：

$$f = \min \sum_{m=1}^{180} D(\mathbf{P}_{\theta_m}^*, \mathbf{P}_{\theta_m}) \quad (8)$$

其中， $D(\mathbf{P}_{\theta_m}^*, \mathbf{P}_{\theta_m}) = \sqrt{\sum_{i=1}^{512} |P_{\theta_m, i}^* - P_{\theta_m, i}|^2}$ 。则，

$$f = \min \sum_{m=1}^{180} \sqrt{\sum_{i=1}^{512} |P_{\theta_m,i}^* - P_{\theta_m,i}|^2} \quad (9)$$

## 2) 约束条件

为保证模型的准确性，同时简化算法求解的过程，给定约束条件：CT 系统只可能单向旋转：

$$\theta_i < \theta_j, 0 < i < j < 180 \quad (10)$$

具体地，

$$\begin{aligned} \theta_2 - \theta_1 &> 0 \\ \theta_3 - \theta_2 &> 0 \\ &\dots \\ \theta_{180} - \theta_{179} &> 0 \end{aligned}$$

即：

$$\begin{aligned} &\begin{bmatrix} -1 & 1 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & -1 & 1 \end{bmatrix} [\theta_1 \quad \theta_2 \quad \dots \quad \theta_{180}]^T > 0 \\ \text{记 } A = &\begin{bmatrix} -1 & 1 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & -1 & 1 \end{bmatrix}, \boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_{180}]^T, \text{ 则} \\ &A\boldsymbol{\theta} > 0 \end{aligned} \quad (11)$$

## 3) 优化模型

综上所述，针对 CT 系统各参数的标定问题建立优化模型如下：

$$\begin{aligned} f = \min \sum_{m=1}^{180} \sqrt{\sum_{i=1}^{512} |P_{\theta_m,i}^* - P_{\theta_m,i}|^2} \\ \text{s. t. } A\boldsymbol{\theta} > 0 \end{aligned} \quad (12)$$

其中待优化的参数为旋转角度 $\theta_m$ 、旋转中心坐标 $(x_0, y_0)$ 、探测器单元间距 $d_n$ 、基准线到探测器左端的距离 $d_0$ 、比例系数 $k$ 。

### 4.2.3 优化模型的求解

#### 1) 遗传算法

本文选取遗传算法求解 4.2.2 中所建立的优化模型。遗传算法(Genetic Algorithm)由 J. Holland 教授提出，是一种通过模拟达尔文生物进化论以及孟德尔遗传定律的自然进化过程来搜索最优解的一种智能算法<sup>[11]</sup>。它通过对初值进行编码、解码、交配、突变、倒位等变换，经过多代的遗传，以启发式的随机搜索找到最优解。

#### 2) 初始种群的确定

选取遗传算法进行求解，需要选定初始种群。为了减小计算的负担，有利于迭代求解，收敛于最优值，初始种群的数值需要在一个合理范围之内<sup>[12]</sup>。因此，下面通过对附件 2 中的原始数据进行分析，粗略地研究一些数据特点，计算 CT 系统旋转中心坐标 $(x_0, y_0)$ 、探测器单元间距 $d_n$ 、基准线到探测器左端的距离为 $d_0$ 等参数的近似值。

(1) 探测器单元间距 $d_n$ 的近似值计算

利用 Excel2016 软件对附件 2 进行处理，将附件 2 中为 0 的单元格标记，得到下图：

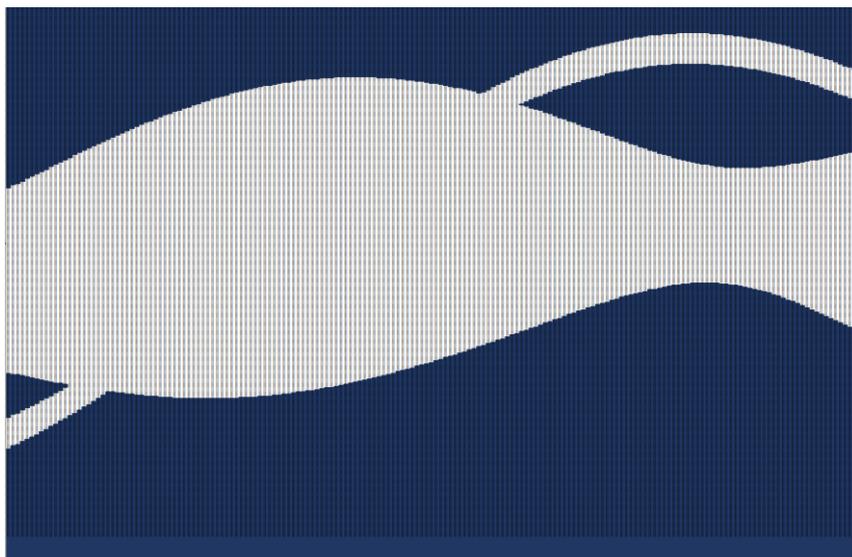


图 4.3 CT 系统扫描待测物件接收数据分布情况

其中，深色部分读数均为 0，即无待测物体的部分；浅色部分均有非零读数，即穿过待测物体的部分。进一步观察可知：

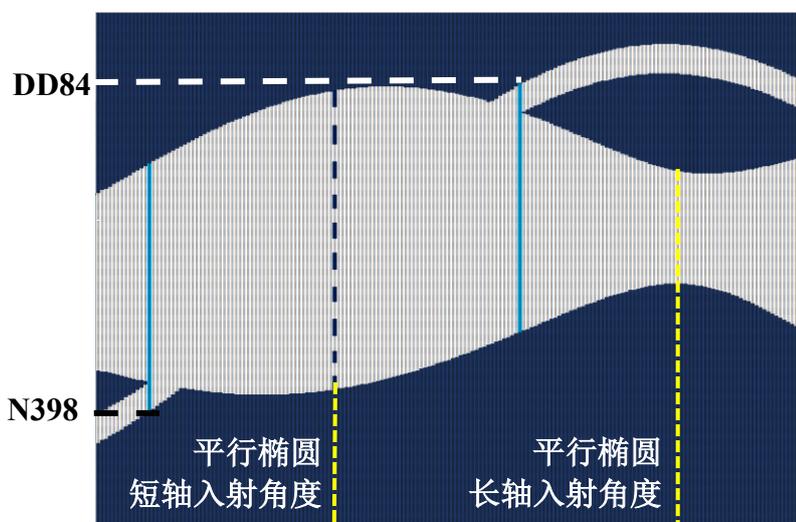


图 4.4 CT 系统扫描待测物件接收数据分析

从图 4.4 中可看出，虚线角度时，投影长度最长覆盖 289 个探测器单元，对应的即为垂直于椭圆长轴的方向。因而可算出探测器单元间隔为  $80/289$  毫米，即  $d_n = 0.2768$  毫米。

(2) 旋转中心及基准线到探测器左端距离的近似值计算

进一步地，在图 4.4 中竖直虚线所对应的角度时，第 235 个单元格即第 235 个探测器单元取得最值，可近似看作过椭圆短轴的射线对应的探测器单元。则知在 X 射线从左水平向右发出时，第 235 条射线大约以椭圆待测物短轴所在直线的路径射入。则，该射线以上有 235 个接收器，以下有 277 个接收器，如下如所示：

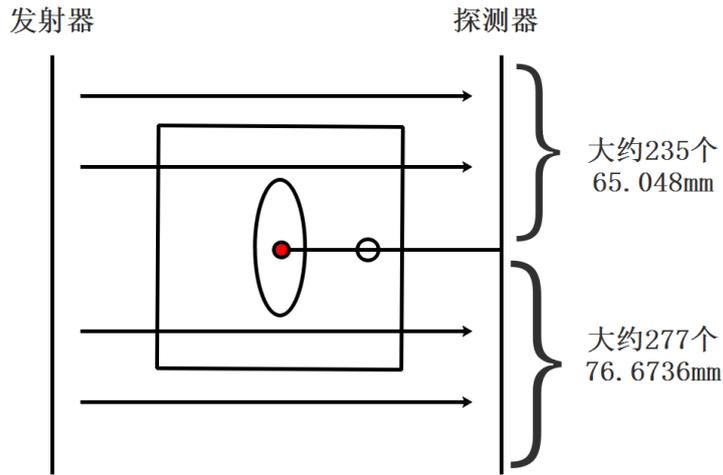


图 4.5 探测器单元分布示意图

进而可计算出探测器各段的尺寸如图 4.5。

因此，此时该两竖直实线代表着待测圆形物体投影刚好完全被投影到探测器上的时刻。由运动是相对的可知，不妨假定探测器不转动，相对地，正方形托盘转动。则，第 14 个角度和第 108 个角度对应的状态如下图：

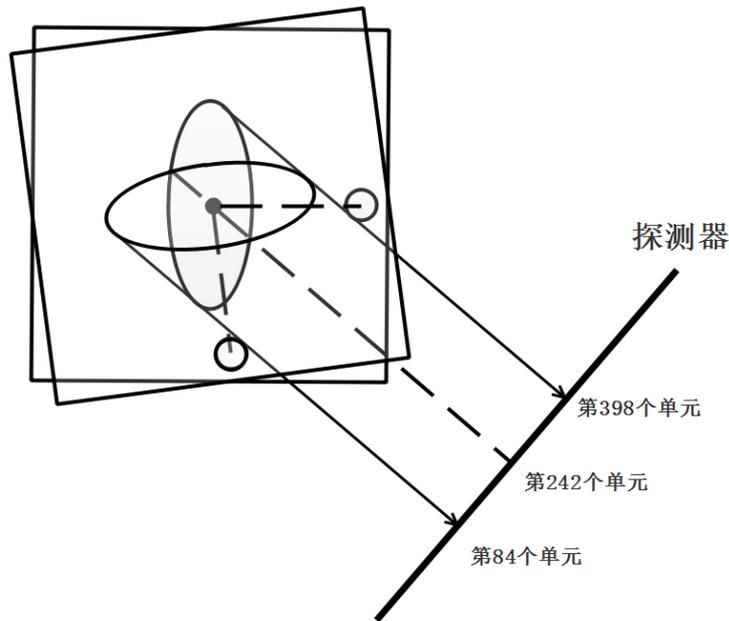


图 4.6 正方形托盘相对旋转示意图

由图 4.6 可知，在第 14 个角度和第 108 个角度时，椭圆待测物与圆形待测物的切线对应的大约是第 84 和第 398 个探测单元。由简单的初等平面几何知识易证，旋转中心对应的是上述两单元格中心，即第 242 个单元格。综上可以粗略计算出旋转中心坐标近似为 $(46.4016, 51.9376)$ ，基准线到探测器左端距离 $d_0 = 66.9858\text{mm}$ 。

### (3) 比例系数 $k$ 的近似值计算

对于比例系数 $k$ ，由图 4.5 可知，在第 151 个角度时，是由平行于椭圆长轴方向入射。则，在该列数据中，第 233 个探测器接收到了读数最大值为 141.7794。又已知长轴长为 80mm，且物体在  $256 \times 256$  的单元格中，则长轴对应的单元格数量为 $80 \cdot (\frac{256}{100})$ 个，而附件 1 中，每个单元格的吸收系数为 1，则比例系数 $k$ 的计算如下：

$$k = \frac{141.7794}{80 \cdot \left(\frac{256}{100}\right)} = 0.6923 \quad (13)$$

可将上述粗略计算的近似值作为遗传算法的初代种群值，带入遗传算法中，以提高算法的准确度。

### 3) 适应度函数及其他基本参数的设定

在遗传算法中，适应度函数用于评价算法的准确度，在此不妨直接以目标函数作为适应度函数，以衡量算法的优越程度。

对于其他的参数，设定如下表：

表 4.1 遗传算法参数取值表

参数名	参数值
种群规模	10
种群代数	200

其他参数均为 MATLAB2017 软件自带 GA 函数的默认值。综合上文，编写遗传算法求解代码<sup>[13]</sup>，详见附录 1。

### 4) 模型求解结果

在求解模型的过程中，出于对算法时间效率和模型求解的精度考虑，本文采用对各方向角度逐个进行优化的方式，求解最优值。最终得到 180 各角度下待优化参数的最优解。对于除角度以外的值，取 180 个优化结果的平均值以保证精度。

求解所得的 180 个方向角度对应最优值详见附件一，取平均值，旋转中心的坐标为：

$$(x_0, y_0) = (46.6736, 54.5797)$$

探测器单元间距为：

$$d_n = 0.2765\text{mm}$$

CT 系统使用 X 射线的 180 个方向角度部分如下表，完整数据见附录：

表 4.2 180 个方向角度求解结果列表

序号	$\theta$								
1	-60.2967	37	-24.2350	73	11.6747	109	47.5138	145	83.2617
2	-58.9435	38	-23.2202	74	12.7115	110	48.5060	146	84.1805
3	-58.4254	39	-22.2225	75	13.7154	111	49.5190	147	85.1118
...	...	...	...	...	...	...	...	...	...
34	-27.2949	70	8.7032	106	44.4887	142	80.3892	178	116.6488
35	-26.2543	71	9.7148	107	45.5191	143	81.5161	179	117.6440
36	-25.2763	72	10.5256	108	46.5214	144	82.4647	180	118.5965

### 4.3 问题二：附件 3 未知介质的图像重建

针对未知介质图像重建的问题，在考虑到本题目中 X 射线影像的采样角度不完全等距，图像信息离散化的像素点，可以采用离散的迭代重建算法(SART)来构建模型。之后，可尝试对模型求解，分析算法复杂度，并进一步考虑如滤波反投影算法等其他算法重建模型，最后利用 MATLAB2017a 软件编程求解，得到附件三的重建图像及相关信息。思路示意图如下：

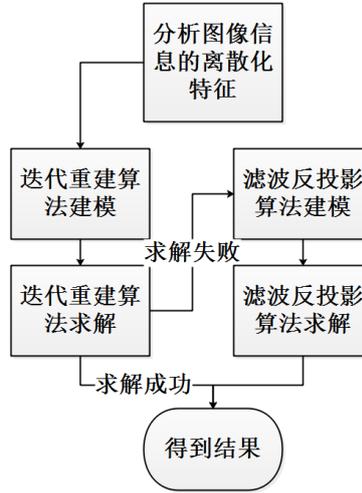


图 4.7 问题 2 的求解思路示意图

#### 4.3.1 基于 SART 算法的图像重建模型

##### 1) SART 算法的理论基础及图像向量模型的建立

考虑到本问题未知介质的信息以矩阵形式给出，可将图像看作离散像素点，使用单下标表示待重建图像的像素按顺序，得到图像向量，并赋予初值<sup>[14]</sup>。对于本文所研究的问题，其表达式如下：

$$\overrightarrow{x^{(0)}} = (x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)})^T \quad (N = 256^2) \quad (14)$$

在给定初值后，将所有的 X 射线所得的投影数值也记为一个向量 $\vec{p}$ ，即：

$$\vec{p} = (p_1, p_2, \dots, p_M)^T \quad (M = 512 \times 180) \quad (15)$$

建立反映各个像素对各条射线的贡献矩阵 $A_{M \times N}$ ：

$$A_{M \times N} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{M,1} & \cdots & a_{M,N} \end{bmatrix} \quad (16)$$

这一贡献矩阵 $A_{M \times N}$ 可以采用点模型来确定<sup>[14]</sup>，即将 X 射线看作有宽度的线，将各个像素看作没有大小的点，且位于每个格子的中央。对于第 $i$ 条 X 射线，它通过了某个像素点 $x_j^{(0)}$ ，则 $a_{i,j} = 1$ ；反之则 $a_{i,j} = 0$ 。则，其计算投影值应为：

$$p_i = k \cdot \sum_{j=1}^N a_{i,j} x_j \quad (17)$$

其中 $k$ 为比例系数。

##### 2) SART 算法求解模型的一般过程

迭代重建算法(Simultaneous Algebraic Reconstruction Technique)是一种代数算法，

适用于求解上述的离散型问题，其求解过程如下：在初值 $\overrightarrow{x^{(0)}}$ 的基础上，对于每个像素点，遍历计算所有射线理论投影值 $p_i$ 与实际投影值 $p_i$ 的差，考虑一条射线的差值，对 $\overrightarrow{x^{(0)}}$ 进行修正。重复此步骤，在 $\overrightarrow{x^{(n)}}$ 得到 $\overrightarrow{x^{(n+1)}}$ ，得到越来越接近实际物体的重建图像。在郭静钰的研究中，SART 算法相比代数重建法(Algebraic Reconstruction Technique)、最大似然期望最大算法(ML-EM)等算法重建的图像有着更高的清晰度<sup>[8]</sup>。

建立基于 SART 算法的模型重建图像，其模型的表达式为：

$$x_j^{(n+1)} = x_j^{(n)} - \delta \times \frac{a_{i,j} \times \sum_{i=1}^M \left( \frac{a_{i,j} \times (p_i' - p_i)}{\sum_{j=1}^N a_{i,j}} \right)}{\sum_{i=1}^M a_{i,j}} \quad (18)$$

其中 $\delta$ 为反映收敛快慢的系数。设定初始向量 $\overrightarrow{x^{(0)}} = (0, 0, \dots, 0)^T$ ，使用 MATLAB 2017a 软件对附件三的投影进行求解。对于每一次迭代，程序遍历所有的射线，并求出每个像素点的修正量。程序代码见附录 3。

### 3) SART 算法求解本问题的不可行性说明

在进行编程求解上述模型的过程中，建立的 $A_{M \times N}$ 贡献矩阵元素数量为 $6.04 \times 10^9$ 个，其占用存储空间高达 40GB，在有限的时间内，不足以得到有效的结果。可见，SART 算法时空复杂度较高，实用的便捷性有待提高。考虑到时间的限制，以及本问题对模型重建的图像清晰度无过高要求，故寻求其他复杂度较低、实现更为快捷的算法模型。

### 4.3.2 基于滤波反投影算法的图像重建模型建立与求解

考虑到本问题对成像清晰度要求相对较低，对运算速度的要求较高，本文选用运算速度快、方法成熟的滤波反投影算法(FBP Method)建立图像重建模型，对附件三进行图像重建。

#### 1) 滤波反投影算法的理论基础

滤波反投影算法(Filtered back projection)也称累加法，其原理在于将断层平面中的某点密度值看作该平面内所有经过该点的射线投影之和<sup>[15]</sup>。FBP 的直接理论基础是中心切片定：一个二元函数 $f(x, y)$ 在某个方向上的投影 $\rho(t)$ 的一维傅里叶变换等于二元函数 $f(x, y)$ 的二维傅里叶变换在这个投影方向上的剖面<sup>[16]</sup>。若记 $\rho(t)$ 的傅里叶变换为：

$$F(\omega) = \int_{-\infty}^{+\infty} \rho(t) e^{-i \times 2\pi \omega t} dt \quad (19)$$

$f(x, y)$ 的二维傅里叶变换为：

$$F(u, v) = \int_{-\infty}^{+\infty} dx \int_{-\infty}^{+\infty} f(x, y) e^{-i \times 2\pi (ux + vy)} dy \quad (20)$$

对 $F(x, y)$ 进行极坐标变换得 $F(\omega, \theta)$ ，则中心切片定理可以写作：

$$F(\omega) = \int_{-\infty}^{+\infty} dx \int_{-\infty}^{+\infty} f(x, y) e^{-i \times 2\pi \omega (x \cos(\theta_0) + y \sin(\theta_0))} dy = F(\omega, \theta_0) \quad (21)$$

其中 $\theta_0$ 为投影方向。

对于 $f(x, y)$ 的投影 $\rho(t)$ , 其与原函数 $f(x, y)$ 满足 Radon 变换关系, 即 $\rho(t)$ 为 $f(x, y)$ 在投影线上的线积分:

$$\rho(t) = \int f(x, y) ds \quad (22)$$

对于由投影求原函数的问题, 实际上也就是求得投影的 Radon 逆变换的过程。所以, 若得到了在 $180^\circ$ 范围内的投影数据, 根据中心切片定理, 可以通过积分计算出待求的 $f(x, y)$ 。但是, 如下图所示, 在将各个“切片”累加的过程中, 频域面上的低频区域会过分加和, 影响图像还原效果。

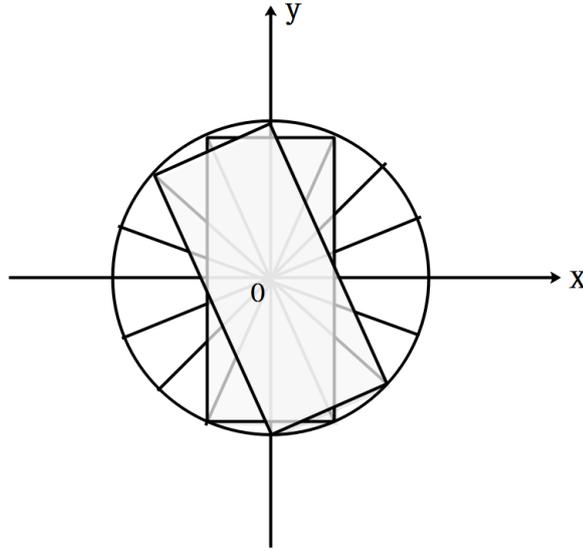


图 4.8 低频过度叠加示意图

为解决此问题, FBP 算法采用了对采样得到的投影进行一维傅里叶变换, 并用滤波函数减弱低频区域的影响的方法<sup>[17]</sup>。其滤波函数一般取为:

$$\mu(\omega) = \begin{cases} |\omega|, & |\omega| \leq \omega_0 \\ 0, & \text{其他} \end{cases} \quad (23)$$

## 2) 基于中心切片定理的反投影模型建立

在上文的理论基础上, 本文建立基于中心切片定理的反投影模型。该模型的建立可以分为两个阶段:

首先, 对方向为 $\theta - \frac{\pi}{2}$ , 即投影角为 $\theta$ 的投影 $\rho_\theta(t)$ 进行傅里叶变换:

$$F_\theta(\omega) = \int_{-\infty}^{+\infty} \rho_\theta(t) \times e^{-i \times 2\pi\omega t} dt \quad (24)$$

对于所得到的频谱函数, 将其与滤波函数相乘, 并进行反变换:

$$R_\theta(t) = \int_{-\infty}^{+\infty} F_\theta(\omega) |\omega| e^{i \times 2\pi\omega t} d\omega \quad (25)$$

之后, 将 180 个这样的 $R_\theta(t)$ 进行积分运算得到待求的重建图像函数 $f_{cal}(x, y)$ :

$$f_{cal}(x, y) = \int_0^\pi R_\theta(x \times \cos(\theta) + y \times \sin(\theta)) d\theta \quad (26)$$

综上，得到如下模型：

$$\begin{cases} f_{cal}(x, y) = \int_0^\pi R_\theta(x \times \cos(\theta) + y \times \sin(\theta)) d\theta \\ F_\theta(\omega) = \int_{-\infty}^{+\infty} \rho_\theta(t) \times e^{-i \times 2\pi\omega t} dt \\ R_\theta(t) = \int_{-\infty}^{+\infty} F_\theta(\omega) |\omega| e^{i \times 2\pi\omega t} d\omega \end{cases} \quad (27)$$

### 3) 反投影模型的求解

考虑到题目中实际给出的附件三以矩阵的离散形式呈现， $f_{cal}(x, y)$  也为矩阵形式，可将上述模型中的积分写成和式进行数值计算。对于 (26) 式，其和式形式为：

$$f_{cal}(x, y) = \sum_{i=1}^N R_\theta(x \times \cos(\theta_i) + y \times \sin(\theta_i)) \times (\theta_i - \theta_{i-1}) \quad (28)$$

而对于 (24) 与 (25) 式的傅里叶积分，情形还可以进一步简化。借助 MATLAB 2017a 中集成的快速傅里叶变换函数  $Y = \text{fft}(x, n)$  与快速傅里叶逆变换函数  $Y = \text{ifft}(x)$  可以方便地对附件三的矩阵进行傅里叶变换与逆变换。

基于此算法，借助 MATLAB2017a 软件对附件三的投影数据进行编程求解，程序的源代码见附录 4。得到附件三所对应的图像为：

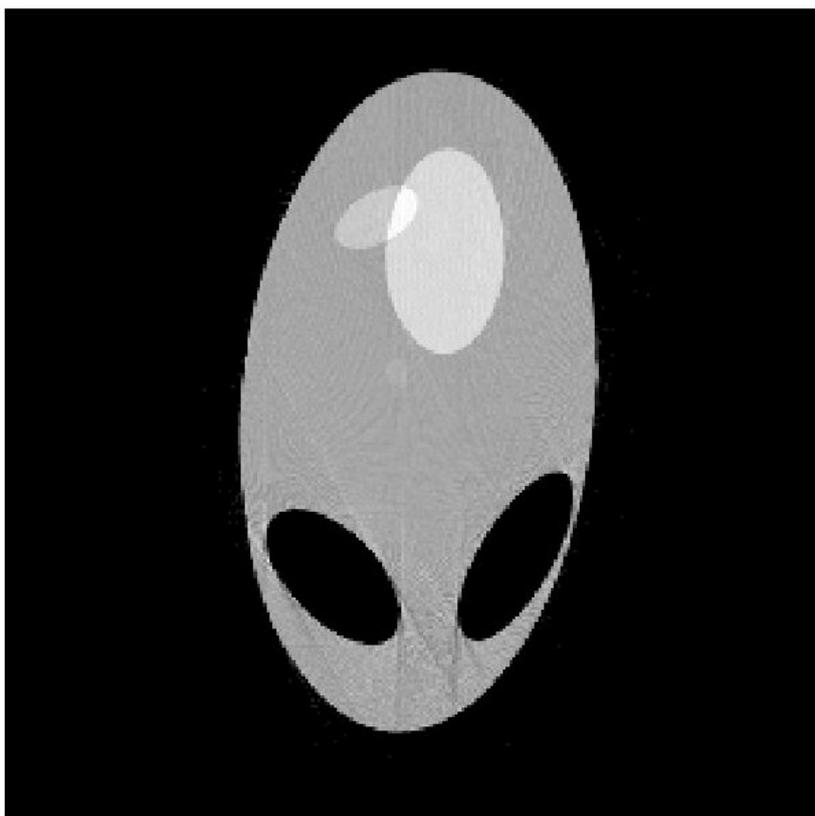


图 4.9 附件 3 未知介质几何形状、位置信息以及吸收率分布图

由图 4.9 即可确定未知介质的位置及几何形状。对于吸收率，定性地，图中较浅的区域表示此处的吸收率较大。完整的吸收强度分布见附件 problem2.xls。对于题目中所给出的 10 个特殊点，所得吸收强度结果见下表：

表 4.3 附件三还原图像中 10 个特殊点吸收强度

横坐标	纵坐标	吸收强度
10.0000	18.0000	0.0000
34.5000	25.0000	0.9957
43.5000	33.0000	0.0000
45.0000	75.5000	1.1999
48.5000	55.5000	1.0393
50.0000	75.5000	1.4658
56.0000	76.5000	1.2805
65.5000	37.0000	0.0000
79.5000	18.0000	0.0000
98.5000	43.5000	0.0000

#### 4.4 问题三：附件 5 未知介质图像信息的确定

对于确定附件 5 中未知介质图像信息的问题，注意到附件 5 与附件 3、附件 2 中的数据形式完全相同，所以理论上可以沿用 4.3 中建立的图像重建模型对问题三进行求解。在尝试使用 4.3 中的方法进行求解后，考虑到附件 5 对应物体结构复杂等特点，可对之前的建模方法进行改进，以得到更精准更清晰的图像结果。

##### 4.4.1 基于反投影模型对附件五的图像重建

借助 4.2.2 中的程序，将附件三换成附件五，即可求得反投影模型对附件五的重建结果，如图 4.10 所示：

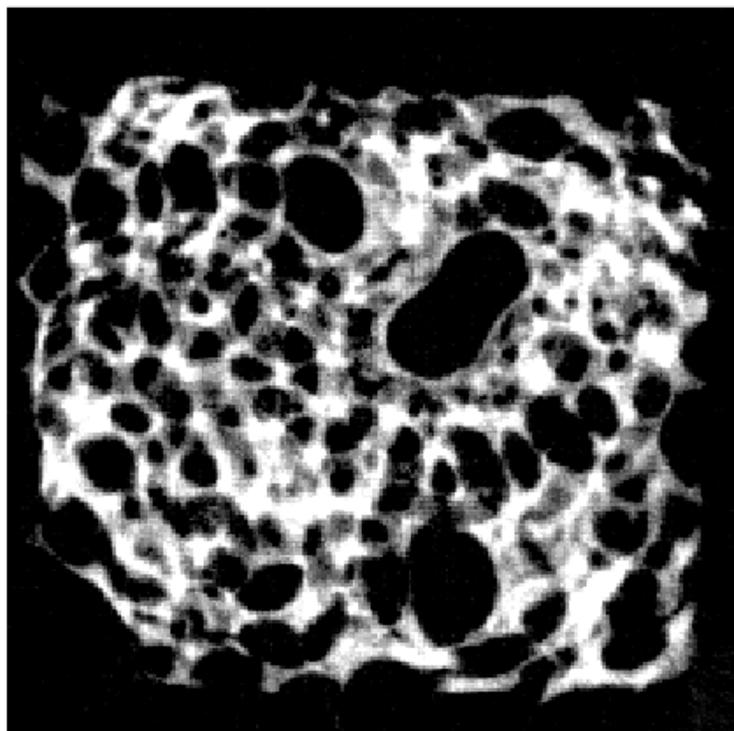


图 4.10 FBP 算法对附件五的重建图像

通过 FBP 算法的求解，得到了附件 5 对应物体吸收率大致的分布图。但是，注意到该图像较为不清晰，且附件 5 对应未知介质几何形状不明显，吸收率分布不规律，结构较为复杂。可见，单纯的 FBP 算法并不能很好地给出附件 5 的重建图像。下面将 FBP 算法进行改进。

#### 4.4.2 基于附件五的复杂图像情况对 FBP 算法的改进

从 4.4.1 的重建结果来看,附件 5 对应的原图像结构形状与吸收率分布较为复杂,而在本问题中的 FBP 算法为离散下的近似,无法保证重建结果的准确性。为解决这一潜在问题,本文参考了一些前人的文献<sup>[18]</sup>,改进了 FBP 模型。在改进后的模型中,重构后的图像将会再次进行 Radon 变换,并将所形成的新投影与原投影比较作差,对原投影进行修正。多次迭代后,重构图像就不断接近真实图像了。改进方法的详细步骤如下:

- a) 对于一个已经计算得到的重建图像函数 $f_0(x, y)$ ,对其再次进行 Radon 变换,得到一个新的投影 $\rho_1(t, \theta)$ :

$$f_0(x, y) \xrightarrow{\text{Radon}} \rho_1(t, \theta)$$

- b) 将新的投影 $\rho_1(t, \theta)$ 与原投影 $\rho_0(t, \theta)$ 作差,得到其差值:

$$\Delta \rho_1(t, \theta) = \rho_0(t, \theta) - \rho_1(t, \theta)$$

- c) 将 $\Delta \rho_1(t, \theta)$ 进行 FBP 变换:

$$\Delta \rho_1(t, \theta) \xrightarrow{\text{FBP}} \Delta f_1(x, y)$$

- d) 傅里叶变换后,所得的 $\Delta f_1(x, y)$ 被用于修正 $f_0(x, y)$ :

$$f_1(x, y) = f_0(x, y) + \Delta f_1(x, y)$$

- e) 对每次 d) 中得出的函数,将其看作计算得到的重建图像函数,带回 a) 中,进行下一次迭代,直至图像满足要求。

其算法示意图如下:

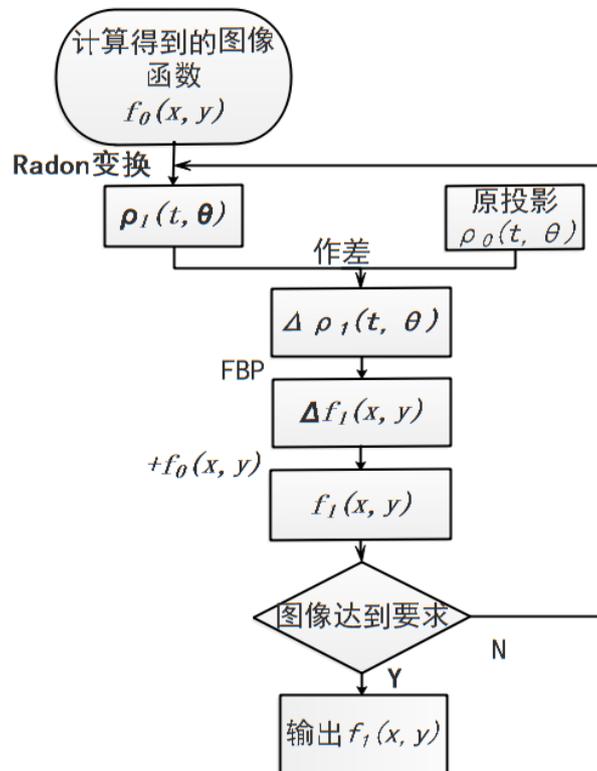


图 4.11 FBP 改进算法流程图

需要指出的是,这种迭代 FBP 算法是在 X 射线转轴过载物台中心即原点的情形下建立的,故在应用于本文时需要因旋转中心偏移而调整重建图像的范围。在第一次 FBP

算法后，若对所得图像的坐标偏移进行修正，则之后每一步所得 $\rho_i(t, \theta)$ 都将是旋转中心准确的情况下所得的投影。这就造成了在作差时，附件5中的实际投影值与 $\rho_i(t, \theta)$ 错位。若不修正，则可能出现部分信息不被包括于 $256 \times 256$ 的图像中。综上，本文将FBP算法所得矩阵增大为 $384 \times 384$ ，即可确保该矩阵涵盖所有有效的信息，且无需逐步地对坐标偏移进行修正。

基于此算法，使用MATLAB2017a进行编程求解，详细代码见附录5。在迭代5次时就已经得到了较为清晰的结果，如下图：

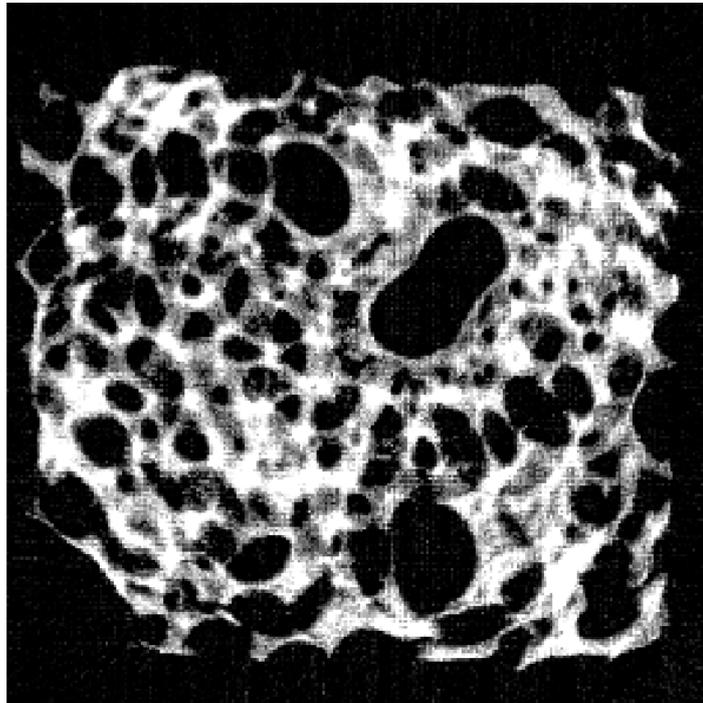


图 4.12 五次迭代 FBP 算法求得的附件五图像

对于指定的 10 个特殊点，所得结果见下表：

表 4.4 附件五还原图像中 10 个特殊点吸收强度

横坐标	纵坐标	吸收强度
10.0000	18.0000	0.0000
34.5000	25.0000	2.6022
43.5000	33.0000	6.5378
45.0000	75.5000	0.0000
48.5000	55.5000	0.1345
50.0000	75.5000	3.1339
56.0000	76.5000	5.9121
65.5000	37.0000	0.1198
79.5000	18.0000	9.1801
98.5000	43.5000	0.1293

完整的吸收强度分布见附件 problem3.xls。

#### 4.5 问题四：参数标定的精准度和稳定性分析及新模板设计

对于分析问题一中参数标定的精度和稳定性以及新模型设计的问题，可结合问题一的数据及求解结果，分析误差和不稳定性的来源，利用相对误差和遗传算法中适应

度函数值分别衡量参数标定的精度和稳定性。基于分析结果，可从精度和稳定性的影响因素出发，建立新的标定模板以及标定模型。

#### 4.5.1 参数标定的精度和稳定性分析

在问题一中，本文通过建立优化模型，利用遗传算法求解，得出了最优的系统参数。在此，选取相对误差反映精度，选取遗传算法中的适应度函数反映稳定性，下面分别加以分析。

##### 1) 精度分析

本文利用实验中常用的多次测量的相对误差来考量参数标定的精度。其表达式如下：

$$E = \frac{\overline{\Delta x}}{\bar{x}} \times 100\% \quad (29)$$

其中，

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \Delta x = \frac{\sum_{i=1}^n |x_i - \bar{x}|}{n}$$

上式中， $n$ 表示数据的个数，在本问题中， $n = 180$ 。

带入上文求解得到的 180 组参数优化求解结果。对于角度的精度分析，本文选取一个角度值进行 10 次优化计算，待其适应度函数值稳定在 5 左右后，计算其相对误差。最后，得到各参数的精度值如下表所示：

表 4.5 各参数的相对误差

参数	中心横坐标 $/x_0$	中心纵坐标 $/y_0$	探测器单元间隔 $/d_n$	比例系数 $/k$	方向角度 $/\theta$
精度	1.4991%	2.1854%	0.1637%	0.1226%	0.0101%

##### 2) 稳定性分析

本文通过分析遗传算法下的 180 组参数的适应度函数值来分析参数标定的稳定性。其 180 个个体的适应度函数值变化如下图所示：

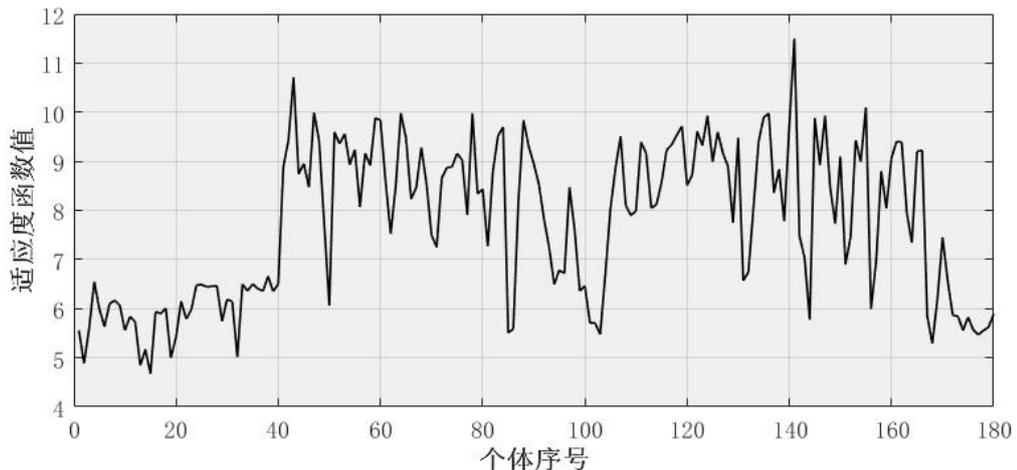


图 4.13 适应度函数变化图像

分析变化曲线可知，其适应度函数值在 7.8 处上下进行小幅度波动。这是通过多次求解取得最优值所致，虽然波动幅度不大，可认为稳定性较优，但仍有改进的空间。

##### 3) 误差及不稳定性来源

由于遗传算法具有一定的随机性，虽然已多次计算取得最优解，仍会产生一定

的随机误差和不稳定性。此外，由于在模拟 CT 计算原理时，模板的图像信息进行了离散化的处理，故其分辨率的大小会对模型的求解结果产生一定影响，带来系统误差和相应的不稳定性。

#### 4.5.2 新模板的设计与各个参数的标定

在 4.2 中的参数标定，虽然其稳定性和精度较好，但其使用了遗传算法这一智能优化算法来确定各个参数的值，不仅耗时较长，还在标定过程中引入了随机因素，造成了一定的不稳定性。接下来本文将改进标定所使用的模板，并建立一套与之配套的标定模型，以在简化算法的基础上保证较高的稳定性与精度。

##### 1) 新模板的设计

考虑分辨率或像素点造成的误差，可以考虑选取如圆形等具有较高对称性的图形作为标定目标，可使任意方向角度 X 射线投影长度相同。为判断旋转方向，本文设计的新模板由两个均质圆柱体组成，其俯视结构如下图所示：

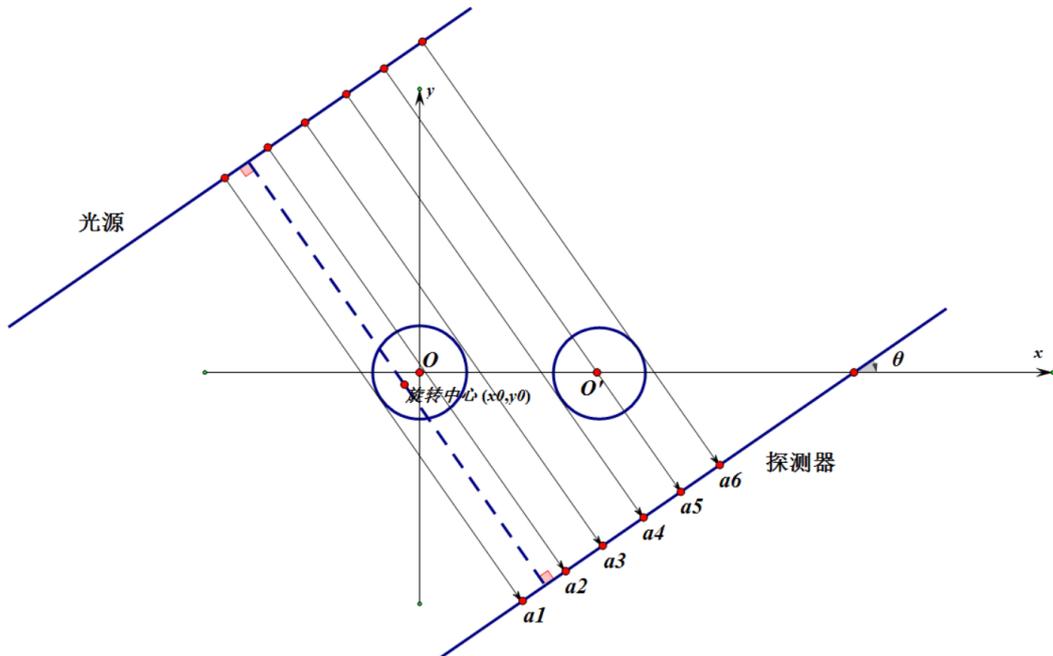


图 4.14 改进的 CT 设备参数标定模板示意图

以正方形托盘的中心点为坐标原点，沿其边长方向建立直角坐标系。其中一个圆柱体的底面圆  $O$  的圆心位于坐标原点，另一个圆柱体的底面圆  $O'$  的圆心在  $x$  轴上。两个圆的半径已知且相等，记其值为  $R$ ；两个圆的圆心距离也事先测得，记为  $l_{OO'}$ 。CT 设备的旋转中心与  $O$  一般不重合。

##### 2) 参数的标定方法

一般来讲，CT 设备待标定的参数有相邻两探测器的间距  $dn$ ，180 个旋转角  $\overrightarrow{\theta_{180}}$  以及旋转中心坐标  $(x_0, y_0)$ 。现探测器绕旋转中心旋转 180 次，得到 180 组投影  $\{\rho_{\theta_i}(t), 1 \leq i \leq 180\}$ 。以下将基于这 180 组投影以及模板的几何结构分别讨论各个参数的标定方法。

首先，对于相邻两个探测器的间距，可任取一组能使两个圆的投影不重叠的投影  $\rho_{\theta_i}(t)$ 。容易得到由于其中一个圆的遮挡，接收端产生信号的探测器个数  $n$ ，则在

不考虑 X 射线的衍射的情况下，相邻两探测器间距可以表示为：

$$d_n = \frac{2 \times R}{n}$$

其次，对于旋转角度的确定，本文在这里不再沿用 4.1 中的定义，改用接收端面与 x 轴的夹角作为旋转角度的标定方法。从图上不难看出，对于每一个  $\rho_{\theta_i}(t)$ ，其  $\theta_i$  满足：

$$|\cos(\theta_i)| = \frac{l_{a_2, a_5}}{l_{OO'}}$$

其中  $a_2$  表示通过原点  $O$  的 X 射线， $a_5$  表示通过圆心  $O$  的 X 射线。考虑到两个圆柱体的投影会有重叠的情况， $a_2$  与  $a_5$  对应的探测器编号  $r_2$  与  $r_5$  的位置可以通过有非零信号的第一个探测器的编号  $r_1$  与最后一个探测器的编号  $r_6$  加以推算，即：

$$l_{a_2, a_5} = l_{a_1, a_6} - 2R = dn \times (r_6 - r_1) - 2R$$

从而有：

$$|\cos(\theta_i)| = \frac{l_{a_1, a_6} - 2R}{l_{OO'}} = \frac{dn \times (r_6 - r_1) - 2R}{l_{OO'}}$$

据此，结合接收端启动与停止的大致位置，即可推算出对应于每个  $\rho_{\theta_i}(t)$  的  $\theta_i$ 。

之后，对于旋转中心位置的标定，需要在得出相邻两探测器的间距  $dn$  与 180 个旋转角  $\overrightarrow{\theta_{180}}$  的前提下进行。基于探测器间距  $dn$  与 180 个角度  $\overrightarrow{\theta_{180}}$ ，方可对每个  $\theta_i$  角度下的投影数据  $\rho_{\theta_i}(t)$  进行图像重建。参照 4.3 中的图像重建方法，若这 180 个角度可以近似看作等距分布，可使用 FBP 算法重建图像；反之，若这 180 个角度无法看作等距分布，或是这 180 个角度分布范围远小于半个圆周，则可使用 SART 算法重建图像。在重建图像的过程中，以旋转中心为原点建立坐标系，则重建所得的图像也为圆心连线平行于 x 轴的两个圆。若其中靠左的那个圆的圆心在当前坐标系中的坐标为  $(x_c, y_c)$ ，则由于在托盘坐标系下这一点为原点，那么当前重建图像坐标系的原点的坐标即为  $(-x_c, -y_c)$ ，也就意味着旋转中心在托盘坐标系下的坐标  $(x_0, y_0) = (-x_c, -y_c)$ 。

### 3) 改进后模板的优点

稳定性上，改进后模板所使用的标定方法单纯地从几何角度考虑问题，并不引入诸如模拟退火、遗传算法这些包含随机性的智能优化算法，具有极好的稳定性。

精度上，原标定方法综合考虑了模板的吸收系数信息，引入了对吸收系数求线积分的 Radon 函数作目标函数进行优化求解。改进后的标定方法则并不直接利用模板的吸收系数信息，对模板的均质性要求不高，也避免了引入含线积分的 Radon 函数，减小了线积分离散化带来的误差。

在标定效率上，原标定方法所采用的遗传算法，其算法复杂度高，运行时间长，往往需要数小时的时间才能优化完成。而改进后的标定方法只使用了 FBP 算法和一些简单的数学计算，对于一幅  $256 \times 256$  像素的图像，FBP 算法往往用数十秒就能求解完成，在运行效率上远高于原标定方法。

## 五、模型的评价与展望

### 5.1 模型的评价

针对 CT 系统参数标定及成像的问题，在旋转中心位置、探测器单元间距、X 射线旋转方向等参数的标定方面，本文首先将问题平面化、解析化，极大地降低了问题分析与模型构建的难度；其次，巧妙地将参数标定转化为优化模型；之后，利用了遗传算法准确地求解了模型。在附件 3 未知介质的位置、形状、吸收率等信息的重建方面，本文先选用了精度较高的迭代重建算法建立图像信息模型，又考虑到其求解的复杂性，从实效性的角度，利用滤波反投影算法建立模型并求解除了未知介质信息。在附件 5 未知介质的相关信息确定方面，本文对滤波反投影算法进行改进，得到了精度更高的吸收率结果与清晰度更高的图像。在分析参数标定精度与稳定性，以及设计新模板和标定模型方面，本文从实验误差和遗传算法中适应度函数两种评估方法出发，找出误差与不稳定性产生的原因，最后以此为基础，设计了改进的简便标定模型。因此，本文所建模型具有以下优点：

- 1) 在 CT 系统参数标定的过程中，将问题自然地以平面几何的角度分析，巧妙地转换成了优化问题，具体地分析数据特征选定了初值，高效地对算法进行了简便求解。
- 2) 在未知介质位置、几何形状和吸收率等信息的确定过程中，对比了不同方法的精确性与时效性，选用精确度高的迭代重建算法建立模型，选用求解效率高的滤波反投影算法进行问题求解，兼顾了理论性与实用性，并对复杂问题中的算法进行了进一步的改进。

但是，由于时间与硬件设备不足，本文无法对基于 SART 算法的图像重建模型进行有效求解，在准确性上遗留了一定的提升空间。同时，由于题目提供的条件与物理知识的限制，本文将 X 射线看作直线传播，并未考虑 x 射线在介质内的散射与衍射，所建模型与实际情况存在一定差异。

### 5.2 改进与展望

在后续的研究中，可改进硬件设备，增加时间投入，进一步优化 SART 算法的复杂度，将基于迭代重建算法的重建模型有效地求解；同时，可结合固体物理、材料科学、波动光学等学科知识，综合考虑 X 射线在介质中的各种性质，使模型更加符合实际。总而言之，CT 技术作为目前广泛使用的无损检测技术，在工业、医疗、安保、考古等领域都有着极高的重要性。建立一套准确稳定而且快速高效的 CT 系统参数标定及影像重建技术，将极大地推动上述领域的发展，改善人们的生产生活。

## 六、参考文献

- [1] 李保磊. X射线CT成像技术进展综述[A]. 中国体视学学会. 第十二届中国体视学与图像分析学术会议论文集[C]. 中国体视学学会, 2008: 6.
- [2] 庄天戈. CT原理与算法[M], 上海交通大学出版社, 1992.
- [3] Ge Wang, Yangbo Ye, Hengyong Yu. Approximate and exact cone beam reconstruction with standard and non-standard spiral scanning[J]. Phys. Med. Bio, 2007, 01(52): R1—R13.
- [4] 西门子推出世界上最先进的CT系统[J]. 中国医疗器械信息 2006.12(10): 80.
- [5] 刘明进. 工业CT系统旋转中心定位方法研究[D]. 重庆大学, 2014.
- [6] 孟凡勇, 李忠传, 杨民, 李静海. CT旋转中心的精确确定方法[J]. 中国体视学与图像分析, 2013, 18(04): 336-341.
- [7] 毛小渊. 二维CT图像重建算法研究[D]. 南昌航空大学, 2016.
- [8] 郭静钰. 有限角度CT图像重建迭代算法研究[D]. 南方医科大学, 2016.
- [9] 钟锡华. 现代光学基础[M]. 北京: 北京大学出版社, 2003.
- [10] 林萌, 李翠华, 黄剑航. 基于Radon变换的运动模糊图像参数估计[J]. 计算机技术与发展, 2008, (01): 33-36.
- [11] (加)Haykin S. 神经网络与机器学习[M]. 申富饶, 徐焯, 郑俊, 晁静译. 北京: 机械工业出版社, 2011.
- [12] 曹春红. 几何约束求解技术的研究[D]. 吉林大学, 2005.
- [13] 卓金武. MATLAB在数学建模中的应用(第二版)[M]. 北京: 北京航空航天大学出版社, 2014.
- [14] 郭立倩. CT系统标定与有限角度CT重建方法的研究[D]. 大连: 大连理工大学, 2016.
- [15] 闫镔, 李磊. CT图像重建算法[M]. 北京: 科学出版社, 2014.
- [16] 马存宝, 周伟. 基于改进二维图像重建算法的工业CT系统[J]. 计算机工程, 2007, 33(17): 219-221.
- [17] 郭琦, 孔斌. 一种基于角度相关性的快速FBP算法[J]. 计算机系统应用, 2008, 17(06): 48-51.
- [18] 祝俊晨, 徐伯庆. 基于FBP的迭代算法研究[J]. 信息技术, 2014(12): 138-140.

## 七、附录

### 附录 1: 优化模型遗传算法求解代码及其适应度函数 (yichuan.m 和 fitness.m)

```
(1) 遗传算法主程序 (文件名: yichuan.m)
%这里所求出旋转角度  $\theta$  减去 180 度与文中对应 !!
%运行时间大约 1 min
%运行时需将 A 题附件放入相同目录中
global t
t=1; %对第 t 个方向的投影数据进行参数优化

x0=46.4010; %旋转中心 x0 初始值
y0=59.4376; %旋转中心 y0 初始值
dn=0.2768; %光源距离初始值
k=0.6923; %比例系数 K 初始值
d0=66.9896; %基准线距第一条 X 射线的距离

chushizhi=zeros(6,10);
theta0=zeros(1,10);
global data1;
global data2;
data1=xlsread('A 题附件.xls',1);
data2=xlsread('A 题附件.xls',2);

lb=[117;46;51.5;61.5;0.26;0.68];
ub=[119+t;48.9;55.7;70;0.29;0.7];

%产生初始种群
for m=1:10
theta0(m)=118+t*0.98+rand()*5;
chushizhi(:,m)=[theta0(m),x0+rand()*0.5,y0+rand()*0.5,d0+rand(),dn+rand()*0.05,k+rand()*0.05]';
end
%chushizhi(:,10)=z; %利用较优个体作为初始种群
chushizhi=chushizhi';
options=gaoptimset('PopulationSize',10,'Generations',200,'InitialPopulation',chushizhi);
[z,val]=ga(@fitness,6,[],[],[],[],lb,ub,[],options);

%这里所求出旋转角度  $\theta$  减去 180 度与文中对应 !!

(2) 适应度函数 (文件名: fitness.m)
function obj=fitness(bianliang)
global data1;
global data2;
```

```

global t;

%bianliang=z; %检验
THETA=bianliang(1);
x0=bianliang(2)/100*256+54;
y0=bianliang(3)/100*256+54;
d0=x0-bianliang(4)/100*256;
dn=bianliang(5)/100*256;
k=bianliang(6);

IMG=data1;
[lt, wt] = size(IMG);
iDiag = sqrt(lt^2 + wt^2);
ld = ceil(iDiag - lt) + 2;
wd = ceil(iDiag - wt) + 2;
padIMG = zeros(lt+ld, wt+wd);
padIMG(ceil(ld/2):(ceil(ld/2)+lt-1), ...
        ceil(wd/2):(ceil(wd/2)+wt-1)) = k*IMG;

n = size(padIMG, 1);
x = linspace(1, 365, n);
[X1, Y1]=meshgrid(x, x);
s=zeros(1, 512);
for i=1:512
    s(i)=d0+(i-1)*dn;
end
[sx, sy]=meshgrid(s, x);
PR = zeros(512, 1);

%for i = 1:180
    i=1;
    theta = (90-THETA(i))*pi/180;
    X = cos(theta)*(sx-x0)+x0+ -sin(theta)*(sy-y0);
    Y = sin(theta)*(sx-x0)+ cos(theta)*(sy-y0)+y0;
    % 二维插值
    tmpimg = interp2(X1, Y1, padIMG, X, Y);
    tmpimg(isnan(tmpimg)) = 0;
    % 累加
    PR(:, i) = (sum(tmpimg))';
%end

pr=PR;
pp=data2(:, t);
obj=norm(pp-pr);

```

附录 2: 180 个方向角度

序号	$\theta/\text{度}$	序号	$\theta/\text{度}$	序号	$\theta/\text{度}$	序号	$\theta/\text{度}$	序号	$\theta/\text{度}$
1	-60.2967	37	-24.2350	73	11.6747	109	47.5138	145	83.2617
2	-58.9435	38	-23.2202	74	12.7115	110	48.5060	146	84.1805
3	-58.4254	39	-22.2225	75	13.7154	111	49.5190	147	85.1118
4	-57.3341	40	-21.2857	76	14.5419	112	50.5208	148	86.2685
5	-56.3043	41	-20.1952	77	15.6462	113	51.4609	149	87.1449
6	-55.3107	42	-19.1463	78	16.8351	114	52.4504	150	88.6246
7	-54.3002	43	-18.1577	79	17.6130	115	53.4494	151	89.9509
8	-53.3078	44	-17.2781	80	18.5806	116	54.3845	152	90.9994
9	-52.3122	45	-16.5567	81	19.6723	117	55.3803	153	92.0000
10	-51.2495	46	-15.2909	82	20.6029	118	56.3509	154	93.0000
11	-50.22	47	-14.1651	83	21.5183	119	57.4295	155	94.0000
12	-49.3297	48	-13.1403	84	22.2612	120	58.4608	156	95.0000
13	-48.3173	49	-12.4825	85	23.6229	121	59.4169	157	96.0000
14	-47.2956	50	-11.3356	86	24.6145	122	60.4275	158	97.0000
15	-46.3161	51	-10.3155	87	25.4120	123	61.3523	159	98.0000
16	-45.0912	52	-9.2958	88	26.3273	124	62.3655	160	98.8263
17	-44.2357	53	-8.2242	89	27.2953	125	63.3932	161	99.8246
18	-43.2116	54	-7.2031	90	28.7691	126	64.3551	162	100.9172
19	-42.2584	55	-6.5694	91	29.7621	127	65.3908	163	101.9085
20	-41.2585	56	-5.2393	92	30.6384	128	66.4252	164	102.7732
21	-40.1874	57	-4.1605	93	31.6379	129	67.4632	165	103.8476
22	-39.0825	58	-3.2054	94	32.6321	130	68.4733	166	104.8412
23	-38.1944	59	-2.1560	95	33.6445	131	69.5159	167	105.8720
24	-37.2177	60	-1.2175	96	34.6421	132	70.5026	168	106.7154
25	-36.2455	61	-0.2833	97	35.7309	133	71.4110	169	107.6914
26	-35.1394	62	0.6962	98	36.7393	134	72.3557	170	108.6776

27	-34.2304	63	1.8200	99	37.7400	135	73.4359	171	109.6540
28	-33.1695	64	2.5516	100	38.7410	136	74.4892	172	110.6817
29	-32.2621	65	3.6171	101	39.6178	137	75.4860	173	111.6492
30	-31.2381	66	4.6285	102	40.6211	138	76.4309	174	112.6334
31	-30.3495	67	5.8428	103	41.6258	139	77.3292	175	113.6086
32	-29.4376	68	6.3163	104	42.5254	140	78.2936	176	114.6067
33	-28.2953	69	7.5960	105	43.5246	141	79.3708	177	115.6106
34	-27.2949	70	8.7032	106	44.4887	142	80.3892	178	116.6488
35	-26.2543	71	9.7148	107	45.5191	143	81.5161	179	117.6440
36	-25.2763	72	10.5256	108	46.5214	144	82.4647	180	118.5965

附录3: SART算法程序 (文件名: SART.m)

```

x0=54.9014;
y0=46.6736;
dn=0.2765;
di=64.0266;
k=0.6973;
theta=reshape([ 29.7033  47.7416  65.7650  83.4306  101.6747  119.7621
137.5138  155.3908  173.2617  191.9085
 31.0565  48.7415  66.7798  84.7607  102.7115  120.6384  138.5059
156.4252  174.1805  192.7732
 31.5746  49.8126  67.7775  85.8395  103.7154  121.6379  139.5190
157.4632  175.1118  193.8476
 32.6659  50.9175  68.7143  86.7946  104.5419  122.6321  140.5209
158.4733  176.2685  194.8412
 33.6957  51.8056  69.8048  87.8440  105.6461  123.6444  141.4609
159.5159  177.1449  195.8720
 34.6893  52.7823  70.8537  88.7825  106.8351  124.6421  142.4504
160.5026  178.6246  196.7154
 35.6998  53.7545  71.8423  89.7167  107.6130  125.7309  143.4494
161.4110  179.9509  197.6914
 36.6922  54.8606  72.7219  90.6962  108.5806  126.7393  144.3845
162.3557  180.9994  198.6776
 37.6878  55.7696  73.4433  91.8200  109.6723  127.7400  145.3803
163.4359  182.0000  199.6540
 38.7505  56.8305  74.7091  92.5516  110.6029  128.7410  146.3509
164.4892  183.0000  200.6817
 39.7800  57.7379  75.8349  93.6171  111.5183  129.6177  147.4295
165.4860  184.0000  201.6492

```

```

40.6703  58.7619  76.8597  94.6285  112.2612  130.6211  148.4608
166.4309 185.0000 202.6334
41.6827  59.6505  77.5175  95.8428  113.6229  131.6258  149.4169
167.3292 186.0000 203.6086
42.7044  60.5624  78.6644  96.3163  114.6145  132.5254  150.4275
168.2936 187.0000 204.6067
43.6839  61.7047  79.6845  97.5960  115.4120  133.5246  151.3523
169.3708 188.0000 205.6106
44.9088  62.7051  80.7042  98.7032  116.3273  134.4887  152.3655
170.3892 188.8263 206.6488
45.7643  63.7457  81.7758  99.7148  117.2953  135.5191  153.3932
171.5161 189.8246 207.6440
46.7884  64.7237  82.7969 100.5256 118.7691  136.5214  154.3551
172.4647 190.9172 208.5965],1,[]);%读入 θ 度数
theta=theta*2.86/180-1.3;
x_start=zeros(256,256);
x_current=x_start;
x_new=x_current;
x_current0=reshape(x_current,1,[]);
x_new0=reshape(x_new,1,[]);
kongzhi=1;
%构建贡献矩阵
for j0=1:1
    for i0=1:512
        d=(i0-1)*dn-di;
        Mi0=zeros(256,256);
        for i=1:256
            for j=1:256
                xp=j*100/256;
                yp=100-100*i/256;
                if abs((1*yp-
tan(theta(j0))*xp+tan(theta(j0))*x0+tan(theta(j0))*d*sin(theta(j0))+d*cos(theta(j
0))-y0)...
/sqrt(1^2+(tan(theta(j0)))^2))<=100/512
                    Mi0(i,j)=1;
                end
            end
        end
        mi0_0=reshape(Mi0,1,[]);
        a(kongzhi,:)=mi0_0;
        kongzhi=kongzhi+1;
    end
end
end

```

```

p0=reshape(fujian6,1,[]);
%迭代过程, diedai为循环变量, 循环100次
for diedai=1:100
    for i=1:1*512
        p_star(i)=sum(a(i,:).*x_current0);
    end
    delta=p_star-p0;
    for j=1:65536
        X(j)=0;
        for i=1:512
            X(j)=X(j)+(a(i,j)*delta(i)/sum(a(i,:)))/sum(a(:,j));
        end
    end
    x_new0=x_current0-X;
    x_current0=x_new0;
end
shuchu=reshape(x_current0,256,[]);

```

#### 附录4: FBP算法程序 (文件名: FBP.m)

```

fujian1=xlsread('A题附件.xls',1);
fujian3=xlsread('A题附件.xls',3);
fujian4=xlsread('A题附件.xls',4);
fujian2=xlsread('A题附件.xls',2);
fujian5=xlsread('A题附件.xls',5);
R=fujian3;
theta=reshape(...
[ 29.7033  47.7416  65.7650  83.4306  101.6747  119.7621  137.5138
155.3908  173.2617  191.9085
  31.0565  48.7415  66.7798  84.7607  102.7115  120.6384  138.5059
156.4252  174.1805  192.7732
  31.5746  49.8126  67.7775  85.8395  103.7154  121.6379  139.5190
157.4632  175.1118  193.8476
  32.6659  50.9175  68.7143  86.7946  104.5419  122.6321  140.5209
158.4733  176.2685  194.8412
  33.6957  51.8056  69.8048  87.8440  105.6461  123.6444  141.4609
159.5159  177.1449  195.8720
  34.6893  52.7823  70.8537  88.7825  106.8351  124.6421  142.4504
160.5026  178.6246  196.7154
  35.6998  53.7545  71.8423  89.7167  107.6130  125.7309  143.4494
161.4110  179.9509  197.6914
  36.6922  54.8606  72.7219  90.6962  108.5806  126.7393  144.3845
162.3557  180.9994  198.6776
  37.6878  55.7696  73.4433  91.8200  109.6723  127.7400  145.3803
163.4359  182.0000  199.6540

```

```

38.7505 56.8305 74.7091 92.5516 110.6029 128.7410 146.3509
164.4892 183.0000 200.6817
39.7800 57.7379 75.8349 93.6171 111.5183 129.6177 147.4295
165.4860 184.0000 201.6492
40.6703 58.7619 76.8597 94.6285 112.2612 130.6211 148.4608
166.4309 185.0000 202.6334
41.6827 59.6505 77.5175 95.8428 113.6229 131.6258 149.4169
167.3292 186.0000 203.6086
42.7044 60.5624 78.6644 96.3163 114.6145 132.5254 150.4275
168.2936 187.0000 204.6067
43.6839 61.7047 79.6845 97.5960 115.4120 133.5246 151.3523
169.3708 188.0000 205.6106
44.9088 62.7051 80.7042 98.7032 116.3273 134.4887 152.3655
170.3892 188.8263 206.6488
45.7643 63.7457 81.7758 99.7148 117.2953 135.5191 153.3932
171.5161 189.8246 207.6440
46.7884 64.7237 82.7969 100.5256 118.7691 136.5214 154.3551
172.4647 190.9172 208.5965],1,[]);%读入θ的度数

```

```

kuandu=2^nextpow2(size(R,1)); bianhuan=fft(R,kuandu);
filter=2*[0:(kuandu/2-1),kuandu/2:-1:1]'/kuandu;
pinlvhanshu=zeros(kuandu,180);
for i=1:180
pinlvhanshu(:,i)=bianhuan(:,i).*filter;
end
fanhanshu=real(ifft(pinlvhanshu));
jieguo=zeros(384,384);
for i=1:180
rad=theta(i)*pi/180;
for x=(-384/2+1):384/2
for y=(-384/2+1):384/2
t=round(x*1.413*cos(rad+pi/2)+y*1.413*sin(rad+pi/2));
if t+round(size(R,1)/2)>=1&&t+round(size(R,1)/2)<=size(fanhanshu,1)
jieguo(x+384/2,y+384/2)=jieguo(x+384/2,y+384/2)+fanhanshu(t+...
round(size(R,1)/2),i);
else
end
end
end
end
jieguo=jieguo/180;
jieguo=jieguo*10/3;%根据附件2的数据对每点的吸收系数进行修正
%滤掉杂点

```

```

for i=1:384
    for j=1:384
        if jieguo(i,j)>=0.1
        else
            jieguo(i,j)=0;
        end
    end
end
end
for i=1:10
    chazhi3=jieguo(79:334,89:344);
z(i)=interp2(chazhi3,fujian4(i,1)*256/100,(100-fujian4(i,2))*256/100,...
'nearest');
end
subplot(1,1,1),imshow(jieguo(79:334,89:344)/1.5)
%原图像作图过于明亮，此处乘以一个小系数以保证图像清晰
%最终结果的 jieguo 矩阵为 384*384，实际的 256*256 矩阵存放于 chazhi3

```

附录5：迭代FBP算法程序（文件名：IAFBP.m）

```

fujian1=xlsread('A题附件.xls',1);
fujian3=xlsread('A题附件.xls',3);
fujian4=xlsread('A题附件.xls',4);
fujian2=xlsread('A题附件.xls',2);
fujian5=xlsread('A题附件.xls',5);
P=fujian1;
R=fujian5;
theta=reshape([ 29.7033  47.7416  65.7650  83.4306  101.6747  119.7621
137.5138  155.3908  173.2617  191.9085
 31.0565  48.7415  66.7798  84.7607  102.7115  120.6384  138.5059
156.4252  174.1805  192.7732
 31.5746  49.8126  67.7775  85.8395  103.7154  121.6379  139.5190
157.4632  175.1118  193.8476
 32.6659  50.9175  68.7143  86.7946  104.5419  122.6321  140.5209
158.4733  176.2685  194.8412
 33.6957  51.8056  69.8048  87.8440  105.6461  123.6444  141.4609
159.5159  177.1449  195.8720
 34.6893  52.7823  70.8537  88.7825  106.8351  124.6421  142.4504
160.5026  178.6246  196.7154
 35.6998  53.7545  71.8423  89.7167  107.6130  125.7309  143.4494
161.4110  179.9509  197.6914
 36.6922  54.8606  72.7219  90.6962  108.5806  126.7393  144.3845
162.3557  180.9994  198.6776
 37.6878  55.7696  73.4433  91.8200  109.6723  127.7400  145.3803
163.4359  182.0000  199.6540
 38.7505  56.8305  74.7091  92.5516  110.6029  128.7410  146.3509

```

```

164.4892 183.0000 200.6817
    39.7800  57.7379  75.8349  93.6171 111.5183 129.6177 147.4295
165.4860 184.0000 201.6492
    40.6703  58.7619  76.8597  94.6285 112.2612 130.6211 148.4608
166.4309 185.0000 202.6334
    41.6827  59.6505  77.5175  95.8428 113.6229 131.6258 149.4169
167.3292 186.0000 203.6086
    42.7044  60.5624  78.6644  96.3163 114.6145 132.5254 150.4275
168.2936 187.0000 204.6067
    43.6839  61.7047  79.6845  97.5960 115.4120 133.5246 151.3523
169.3708 188.0000 205.6106
    44.9088  62.7051  80.7042  98.7032 116.3273 134.4887 152.3655
170.3892 188.8263 206.6488
    45.7643  63.7457  81.7758  99.7148 117.2953 135.5191 153.3932
171.5161 189.8246 207.6440
    46.7884  64.7237  82.7969 100.5256 118.7691 136.5214 154.3551
172.4647 190.9172 208.5965],1,[]);

```

```

kuandu=2^nextpow2(size(R,1)); %为FFT变换指定宽度
%FFT变换
bianhuan=fft(R,kuandu);
filter=2*[0:(kuandu/2-1),kuandu/2:-1:1]'/kuandu;
pinlvhanshu=zeros(kuandu,180);
for i=1:180
pinlvhanshu(:,i)=bianhuan(:,i).*filter;
end
%反变换
fanhanshu=real(ifft(pinlvhanshu));
jieguo=zeros(384,384);
for i=1:180
rad=theta(i)*pi/180;%化为弧度制
for x=(-384/2+1):384/2
for y=(-384/2+1):384/2
t=round(x*1.413*cos(rad+pi/2)+y*1.413*sin(rad+pi/2));
if t+round(size(R,1)/2)>=1&&t+round(size(R,1)/2)<=size(fanhanshu,1)
jieguo(x+384/2,y+384/2)=jieguo(x+384/2,y+384/2)+...
fanhanshu(t+round(size(R,1)/2),i);
else
end
end
end
end
jieguo=jieguo/180;
jieguo1=jieguo;

```

```

shuchu=zeros(384,384);
for i=1:384
    for j=1:384
        if jieguo(i,j)>=0
            shuchu(i,j)=jieguo(i,j);
        else
            end
    end
end
end

for xunhuan=1:5
    A=radon(shuchu,theta,768);
    A0=zeros(768,180);
    for i=1:768
        for j=1:180
            if A(i,j)>=0
                A0=A;
            else
                end
        end
    end
    end
    pd=R-A0(129:640,:);
    kuandu=2^nextpow2(size(pd,1)); bianhuan=fft(pd,kuandu);
    filter=2*[0:(kuandu/2-1),kuandu/2:-1:1]'/kuandu;
    pinlvhanshu=zeros(kuandu,180);
    for i=1:180
        pinlvhanshu(:,i)=bianhuan(:,i).*filter;
    end
    fanhanshu=real(iffthanshu);
    jieguo=zeros(384,384);
    for i=1:180
        rad=theta(i)*pi/180;
        for x=(-384/2+1):384/2
            for y=(-384/2+1):384/2
                t=round(x*1.413*cos(rad+pi/2)+y*1.413*sin(rad+pi/2));
                if t+round(size(R,1)/2)>=1&& t+round(size(R,1)/2)<=size(fanhanshu,1)
                    jieguo(x+384/2,y+384/2)=jieguo(x+384/2,y+384/2)+...
                    fanhanshu(t+round(size(R,1)/2),i);
                else
                    end
            end
        end
    end
end
end

```

```

jieguo=jieguo/180;
%保持每次结果非负
jieguo0=zeros(384,384);
for i=1:384
    for j=1:384
        if jieguo(i,j)>=0
            jieguo0(i,j)=jieguo(i,j);
        else
            end
    end
end

shuchu=jieguo0+shuchu;
end
shuchu=shuchu*100/85;
%滤掉杂点
for i=1:384
    for j=1:384
        if shuchu(i,j)>=0.1
            else
                shuchu(i,j)=0;
            end
    end
end
end
%画出图象，此处除以三也是由于原图像太亮无法看清细节
subplot(1,1,1),imshow(shuchu(79:334,89:344)/3)
%插值
for i=1:10
    chazhi5=shuchu(79:334,89:344);
y(i)=interp2(chazhi5,fujian4(i,1)*256/100,(100-fujian4(i,2))*256/100,'nearest');
end
%最终得到的 256*256 矩阵存放于 chazhi5 中

```