

平行束 CT 系统的参数标定及成像

摘要

CT 系统是利用 X 射线等线束对人体或样品进行扫描，根据被扫描样品的吸收特性进行断层成像，被广泛应用于医学与工业领域。CT 系统安装时往往会产生误差，本文研究的是平行束 CT 系统的参数标定及成像问题。

针对问题一，首先研究了探测器之间的距离，利用小圆在各个方向投影的不变性，从附件二数据中总结出了小圆投影的三种典型情况，建立了探测器间距与小圆直径的不等关系式，在最大误差不超过 1.57% 的情况下，取探测器间距为 0.2808mm 。其次研究了发射-接收系统与标定模板之间的相对运动。假设发射-接收系统固定不动，选取小圆模板为研究对象，运用微元法，建立了运动微分方程，求解得到小圆模板相对探测器平面的运动方程。根据椭圆模板的特性以及微分方程的变化率，从附件二数据中选取了 3 个特殊点：椭圆长轴、短轴方向以及变化率最低点，代入上述方程中求解出未知参数，由此解得旋转中心相对于小圆模板的位置。以托盘几何中心为原点建立一个绝对坐标系，求解得到的旋转中心在绝对坐标系的坐标为 $(-8.1511, 5.4151)$ 。之后利用寻找极值与 prewitt 算子提取边缘的方法，提取出了 180 个小圆的位置，代入方程中解得相应的旋转方向，范围为 $[22.9792^\circ, 210.0095^\circ]$ 。最后，利用问题二的模型对附件二进行图像重建，以此对标定参数进行再次修正，最终将旋转中心坐标修正为 $(-9.2603, 6.2484)$ ，旋转方向范围修正为 $[29.6031^\circ, 208.6031^\circ]$ 。

针对问题二、三，首先利用第一问所求的旋转中心点的坐标，建立了投影直角坐标系和投影极坐标系。利用滤波反变换方法，对原图进行了重建，利用傅立叶变换和卷积滤波处理，消除了星型伪影的影响，利用线性插值和邻近插值解决了坐标系间离散值转换的难题，利用双线性插值将图形放缩到 256×256 规模的图像，由此得到了重建的原图；再将重建的图像进行二值化处理，得到类似于附件 1 的分布矩阵，并单独给出如椭圆中心点等特征点的位置信息，更好标定位置。最后，将图像重建算法视为黑箱系统，研究附件二的输入数据与输出图像灰度值的关系，以附件一为“吸收率”的基准，得出将原始数据作二倍增益处理后，输出图像的灰度值与吸收率实现统一的规律。据此规律，将附件 3 和附件 5 原始数据二倍增益处理后，进行图像重建，即得到吸收率。

针对问题四，首先对问题一的求解进行了误差分析，椭圆自身的特性限制了求解精度的进一步提高。在保留小圆标定模板的基础上，本文设计了正方形以及正六边形两种新型标定模板，可以在多个方向找到特征值，从而建立出超定方程组，从而减小提取特征点的误差影响。如有相应的实验数据，可用遗传算法等智能搜索算法进行寻优求解。

最后，我们对修正后的旋转中心坐标进行了检验。发现修正后的旋转中心在探测器平面上的投影与探测器平面中心仅有 0.702mm 的误差，几乎重合，这个结论支持了本文的模型和结果。同时也求解出探测器平面在安装时相对于托盘几何中心存在 8.5644mm 的误差。

关键词 CT 系统 平行束 参数标定 滤波反投影 标定模板

一 问题重述

CT系统就一种利用X光，对物体的进行扫描，并根据物质对x光的吸收性不同而产生一维投影，并通过反radon变换生成原二维图像，以此来达到对内部结构进行重构的目的，本文所用CT机为平行X光照射，在对侧的相对位置接收器接收穿过后的X光数据，形成一维radon投射，并绕一中心进行旋转成像，将180组512个参测器的数据进行反向radon变换，形成原图信息。

由于制造工艺的限制，CT机往往存在各种误差，如安装误差，旋转中心误差等，使得重构图像的数据偏差较大，因此提出以下问题：

(1)在如图所示的模板中，通过CT机投影得到的数据为附件2。附件1就是该图在方形托盘上的排布，并且该值就是模板物质吸收强度的真实反映，因此，命名为吸收率。请根据以上信息，求得旋转中心，并求得其位置信息，以及探测器的旋转角度的信息。。

(2) 附件3就是一组CT投影数据，请利用已知信息，求出其形状、位置及吸收率信息，将图3所给的10个特殊点的对应的吸收率求出。

(3) 附件5也是一组CT投影数据，请将形状、位置、吸收率求出，也将上述10个特殊点对应的吸收率求出。

(4) 分析问题一中求解精度，并根据所得结论，自主设计新的模板，并分析精度。

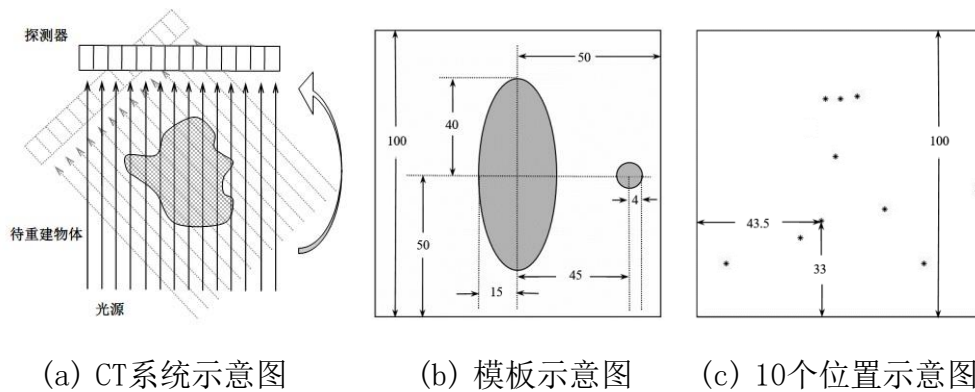


图1 题目给出的三个示意图

二 模型假设

- 1、忽略 X 射线在待测物体中的折射、衍射等现象，假设其均按照直线传播且相邻两源-接收器不互相影响；
- 2、假设所给的标定模板是标准椭圆和圆；
- 3、假设源-接收器运动仅限于水平面，忽略垂直面抖动造成误差；
- 4、假设探测器每次转动的度数为 1 度。
- 5、假设（2）（3）问中所给的附件数据均是在与（1）中完全相同的条件下测得；

三 符号说明

符号	意义	备注
D	标定模板中小圆直径	mm
d_0	探测器单元之间的距离	mm
y	探测器坐标系中平行探测器排列方向	分度值为探测器个数
dy	y 坐标微元	
x	探测器坐标系中垂直探测器排列方向	
R	小圆到旋转中心的距离	单位为探测器个数
θ_0	小圆-旋转中心连线与水平方向夹角	度
θ	小圆-旋转中心连线转过角度	度
$f(x, y)$	待测物体二维分布函数	
$\hat{f}(r, \alpha)$	重建得待测物体二维分布函数	
φ	二维平面内的一条直线 L 与 Y 轴夹角	
s	原点到 L 的垂线距离	
p	实际的射线投影	

四 问题分析

4.1 问题一的分析

问题一是为了标定 CT 系统的安装误差，CT 系统在安装时，理想状态是通过旋转中心的平行光束垂直于探测器平面，且垂足位于探测器平面中心。本问题研究 CT 系统的旋转中心相对于正方形托盘中心的误差。考虑到椭圆相对旋转中心的运动不明显，会增大建模的误差，因此可以研究圆形模板与发射—接收系统的相对运动，假定发射—接收系统固定不动，圆形模板绕发射—接收系统顺时针转动，以第一个探测器为坐标原点，建立了一个探测器坐标系，利用微元法，建立了圆形模板圆心在探测器坐标系里的相对运动微分方程。为了求解方程中的参数，我们对附件二中的数据进行数据处理，选取了三个特征点，分别为平行光束沿椭圆模板短轴和长轴的方向，以及变化率最小的点，将三个点的数据带入微分方程，可以解得微分方程中的三个参数，从而得到了旋转中心与圆形模板的距离，以及两者连线与水平方向的夹角，则可以确定了旋转中心的位置。最后，我们利用在限制窗口内寻找极值和利用 `prewitt` 算子提取边界点的方法，从附件二数据中提取了小圆的 180 个圆心点，将其带入小圆圆心的方程中，得到了 CT 系统使用的 X 射线的 180 个方向。

4.2 问题二、三的分析

问题二是为了在得知旋转中心情况下，将附件 3 的 CT 投影数据，经滤波反雷登变换，将原二维图像进行重建，并表示出该图像在托盘中的位置和吸收率等信息。首先利用第一问以求得的旋转中心建立投影直角坐标系和投影极坐标系。在反投影重建的过程中，容易受到低频杂波的影响，使得重建的图像不清晰，因此，我们采用滤波反投影重建方法，将运用傅立叶变换，卷积滤波，傅立叶切片定理等知识，将 180 角度的投影再二维频率域重组，经过二维反傅立叶变换和插值运算，将图像的矩阵重建，最后，利用二维线性插值，将图像矩阵规模缩放为 256×256 ，由此，就得到了重建图像，并且每一个像素点的位置就是原图在托盘中的位置，为了更为方便的表示位置，我们将图中 6 个椭圆的中心在绝对坐标系下的单位为毫米的坐标值。在吸收率的计算过程中，我们利用黑箱理论分析附件 1 和附件 2 的关系，得到了 CT 投影矩阵与原图吸收率之间的关系，求解出附件 3 重建的图像中每个点的吸收率。

由于问题三和问题二属于同一类问题，在此不再赘述。

4.3 问题四的分析

要设计新的标定模板，首先要分析问题一中标定参数的误差来源。问题一中标定参数是通过选取特殊点，带入方程中求解得到。由于探测器接收的数据是离散的，选取特殊点时会不可避免的产生误差，而受限于椭圆模板的自身特

性，问题一求解时最多选取 3 个特殊点，因此无法进一步提高求解的精度以及对结果进行验证。针对这个问题，可以考虑选用正多边形的标定模版，多边形边数越多，可选择的特殊点就会越多，但同时也会影响实验数据的精确度，因此要综合考虑，确定新的多边形模板。

五 模型的建立与求解

5.1 问题一模型的建立

(1) 探测器之间的距离

利用圆形模板在任何方向上的投影不变的特性，从附件二中的数据可以很分析得到，小圆的直径在不同角度上的投影可以分为 3 种情况，我们选取了第 1、6 和 11 列的数据，此时小圆的投影是与椭圆投影分离的，这 3 组数据可以代表小圆在不同角度的投影情况：

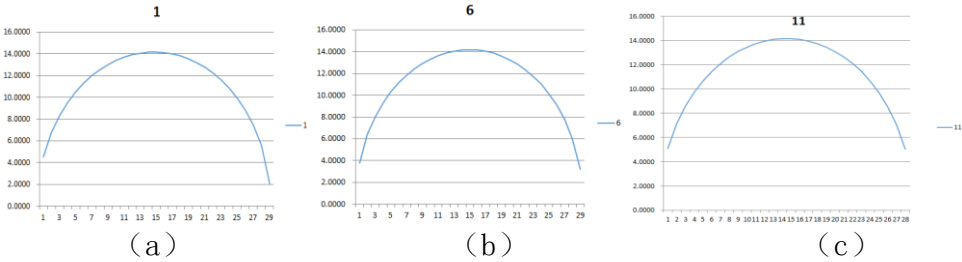


图 2 小圆在不同方向的投影情况

其中，（a）图有 29 个探测器接收到信号，且明显不对称，说明中心线不过圆心；（b）图也有 29 个探测器接收到信号，且成近似对称，我们可以近似认为中心线过圆心；（c）图只有 28 个探测器接收到信号，且成近似对称。根据以上这三种情况，可以研究小圆直径 D 与探测器之间距离的关系。

先研究特殊的情况，（b）图：

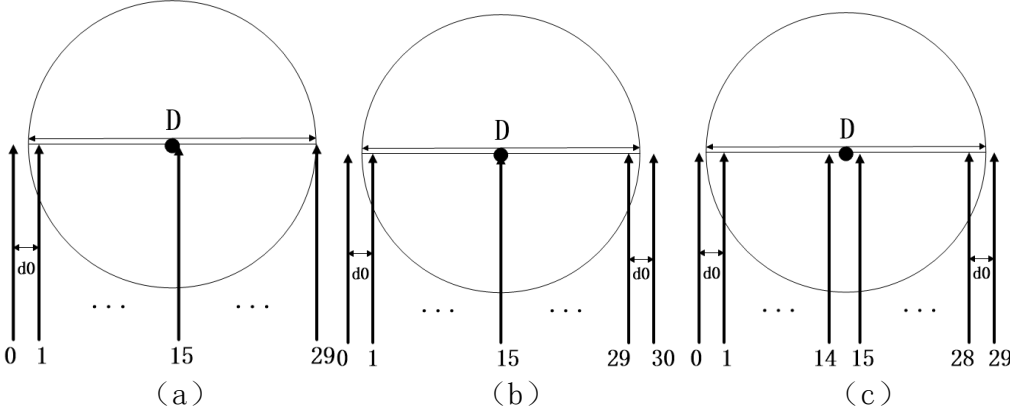


图 3 小圆直径 D 与探测器之间距离的 3 种关系

当处于（a）情况，此时图像显然不对称，但不失一般性，考虑 29 号射线无限接近于相切，可得到不等式：

$$D - 28d_0 < d_0$$

$$D > 28d_0$$

当处于（b）情况，近似认为此时图像是对称的，可以得到不等式：

$$\frac{D - 28d_0}{2} < d_0$$

$$D > 28d_0$$

当处于（b）情况，近似认为此时图像是对称的，可以得到不等式：

$$\frac{D - 27d_0}{2} < d_0$$

$$D > 27d_0$$

由此，可以得到探测器之间的距离和小圆直径的不等关系式：

$$\frac{D}{29} < d_0 < \frac{D}{28} \quad (1)$$

（2）小圆与发射—接收系统的相对运动

题设为发射—接收系统绕一旋转中心逆时针旋转，为了更好地研究探测器与标定模板之间的相对运动规律，我们假设发射—接收系统是固定不动的，正方形托盘及其上的模板绕旋转中心顺时针转动，选择初始时刻如图 4：

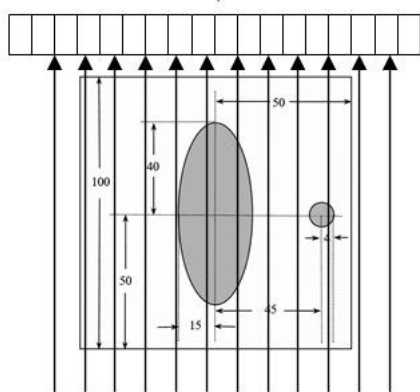


图 4 初始位置示意图

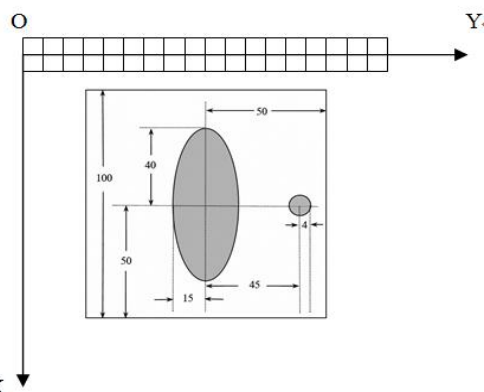


图 5 探测器坐标系示意图

取第一个探测器为坐标原点，建立探测器坐标系，如图 5 所示。

由于旋转中心接近于椭圆的中心，因此椭圆的相对运动不是很明显，研究起来容易引起较大误差。因此，我们选择小圆为研究对象，研究小圆与发射—接收系统的相对运动规律。任意时刻，小圆由初始时刻转动 θ 角的运动规律可由如图 6 所示：

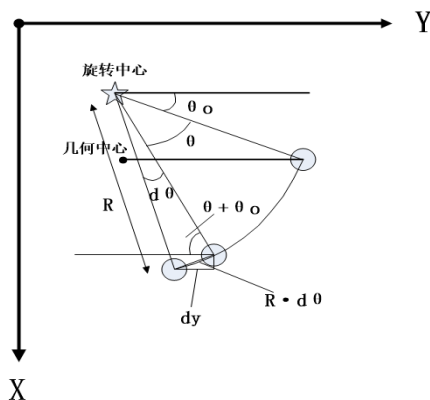


图 6 小圆运动规律示意图

利用微元法，得到小圆运动的微分方程为：

$$dy = -r * \sin(\theta + \theta_0) d\theta \quad (2)$$

不定积分求解得到小圆圆心相对于探测器平面的运动规律：

$$y = r * \cos(\theta + \theta_0) + C \quad (3)$$

其中， r 为小圆到旋转中心的距离， θ_0 为两者连线与水平方向夹角。据此，可以确定旋转中心在正方形托盘中的位置。

(3) 提取小圆圆心位置坐标

为了得到 CT 系统使用的 X 射线的 180 个方向，我们从原数据图像中提取出每个方向下小圆圆心在探测器坐标系中的坐标，即上式中的 y 值，将提取出的 180 个 y 值，带入运动方程，就可以得到每个方向相对于初始位置的旋转角度 θ_i 。

附件二中数据的二维图像如图 7 所示：

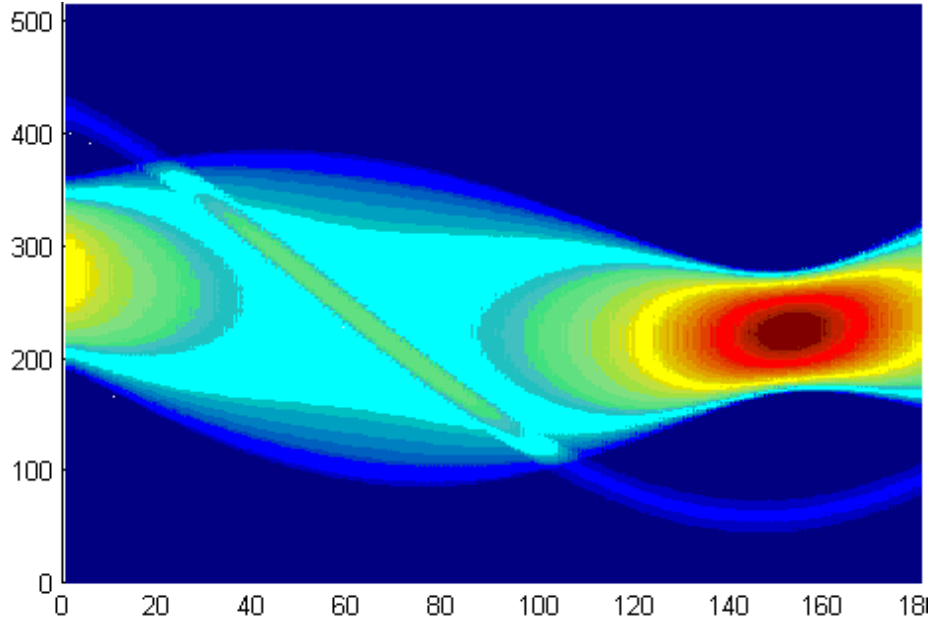


图 7 附件二数据二维图像

从图中可以明显地看出小圆与椭圆的位置关系可以分为两段：分离段和重合段。

1) 分离段

对于分离段，由于小圆直径的方向是最大的衰减方向，因此我们通过限制窗口内寻找极值的方法，可以依此快速地输出每一个方向上圆心点的 y 值。

2) 重合段

对于重合段，由于椭圆和小圆的衰减产生了复合作用，极值点不一定出现在小圆圆心，且小圆的边界也不清晰。因此，我们利用 MATLAB，对图像进行边缘提取。采用的边缘检测算子为 prewitt 算子。

用边缘算子提取出边缘点集之后，将每一列的边缘点求和取平均值，即两条边界的中间值，作为分离段小圆的圆心位置。

将提取出的小圆圆心位置坐标带入方程 (2) 中，即可得到 CT 系统旋转的 180 个方向：

$$\theta_i = \arccos\left(\frac{y_i - C}{r}\right) + \theta_0 \quad (4)$$

5.2 问题二模型的建立

5.2.1.理论基础

(1) 雷登变换及其逆变换

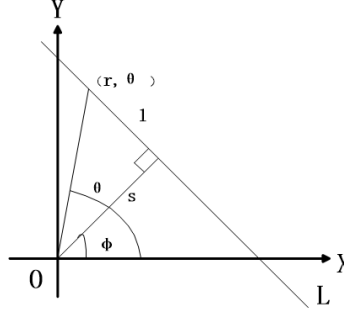


图 8 雷登变换参数示意图

如图 16 所示，二维平面内的一条直线 L 与 X 轴夹角为 φ ，原点到 L 的垂线距离为 s，直线上的点 (x, y) 可以用极坐标表示为 (r, θ) 。

若已知函数 $f(x, y) = \hat{f}(r, \theta)$ 沿直线 L 的线积分为：

$$p = \int_L f(x, y) dl = \int_L \hat{f}(r, \theta) = \int_{-\infty}^{+\infty} \hat{f}(\sqrt{s^2 + l^2}, \varphi + \arctan \frac{l}{s}) dl \quad (5)$$

则：

$$\hat{f}(r, \theta) = \frac{1}{2\pi^2} \int_0^\pi \int_{-\infty}^{+\infty} \frac{1}{r \cos(\theta - \varphi) - s} \frac{\partial p}{\partial s} ds d\varphi \quad (6)$$

式 (4) 称为雷登变换，是指二维分布函数在一定角度下的线积分，即实际的射线投影 p，式 (5) 称为雷登反变换，他指的是通过一定量的投影采样角度下的投影数据 p 可以重建出物体的断层图像 $\hat{f}(r, \theta)$ 。

(2) 反投影重建算法

反投影重建算法的原理是：将断层平面中的某一点的密度值看作这一平面内所有经过该点的射线投影之和。反投影重建算法的任务是寻找经过某个像素点的所有射线投影，将每个射线投影沿射线方向均匀地分配投影值，根据反投影结果映射生成重建图像。

反投影的点扩散函数：

对于 CT 成像系统，一个理想点状物体的图像都会扩散成一个分布，这个分布称为点扩散函数。

反投影重建算法的重建过程可以看作一个从输入原始图像到输出重建图像的成像系统，如下图所示：

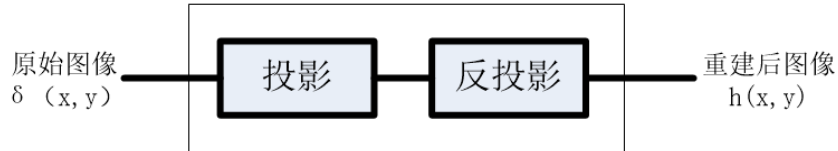


图 9 反投影重建的等效成像系统

该成像系统的点扩散函数为：

$$h(r, \theta) = \frac{1}{\pi} \frac{1}{|r|} = \frac{1}{\pi \sqrt{x^2 + y^2}} \quad (7)$$

从上式可以得出，对应于反投影算法的系统中，点扩散函数并不是 δ 函数，该成像系统是不完美的。在 $r \neq 0$ 处，像素值已经不等于 0。这定量地描述了反投影重建算法星状伪影的本质。

(3) 滤波处理

为了消除星状伪影，在反投影重建之前将投影数据进行滤波处理，再进行反投影重建。过程如下图所示：

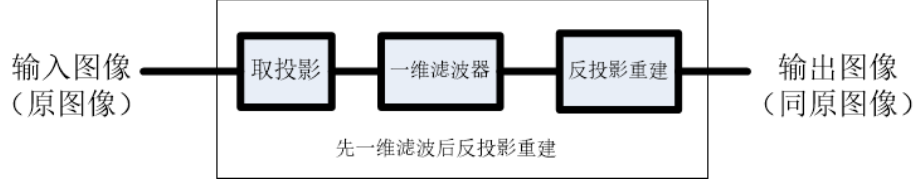


图 10 去除星状伪影的方法

(4) 傅里叶中心切片定理

假设 $f(x, y)$ 为待重建物体的密度函数， $p_\varphi(x_r)$ 为 $f(x, y)$ 在角度 $\varphi = \varphi_0$ 时的平行束投影，有傅里叶中心切片定理的数学表达式为：

$$F_l[p_\varphi(x_r)] = F(\rho, \varphi)|_{\varphi=\varphi_0} \quad (8)$$

其中， $F_l[]$ 表示一维傅里叶变换， $F(\rho, \varphi)$ 是二维傅里叶变换的极坐标表示。

傅里叶切片定理的意义在于，通过在投影上执行傅里叶变换，我们可以从每个投影中得到物体的二维傅里叶变换。

(5) 卷积滤波

由傅里叶变换的卷积原理可知：空域中两个函数的卷积等价于频域内的两个函数的乘积。常用的滤波器有 R-L、S-L、Cosine、Hamming 和 Hanning 滤波器等。本文为了比较验证，选用了 R-L、S-L 和 Hamming3 种滤波器。

(6) 插值方法

在平行束投影的采集过程中，射线的平移采样和角度采样都是离散化的，像素点在 x_r 轴上的投影并不一定位于平移采样点上。因此，反投影过程中内插运算是非常重要的步骤。常用插值算法有紧邻内插和线性内插两种方式。

5.2.2 滤波反投影算法的设计

1、滤波

(1) 滤波的设计：由投影矩阵 P 和阈值 d ，根据滤波的要求，从以上几种滤波函数中选则一种，并生成滤波向量 H 。

(2) 滤波过程的实现

①对投影矩阵的每一个 θ 角度下的 P_f 进行快速傅立叶变换得，由空域转化到频域，简化问题；

$$P_{f, \theta} = \text{Fourier} (P_\theta) \quad (9)$$

②在频域中，乘积运算即为卷积运算，变换后的投影与滤波向量 H 卷积为：

$$P_{h, \theta} = P_{f, \theta} \times H \quad (10)$$

③最后根据傅立叶切片原理组合到二维傅立叶频域，随后进行傅立叶反变换即可求得原矩阵

$$I = i\text{Fourier} (P_{h, \theta}) \quad (11)$$

2、反投影

(1) 投影直角坐标系和投影极坐标系的建立

①以第一问求得的旋转中心为原点，以与托盘下边界平行向右方向为 X 轴，纵向为 Y 轴建立平面直角坐标系

②以旋转中心为极点，以投影直角坐标系夹角为 θ 的方向为极轴，建立投影极坐标系。

坐标系如图 11 所示

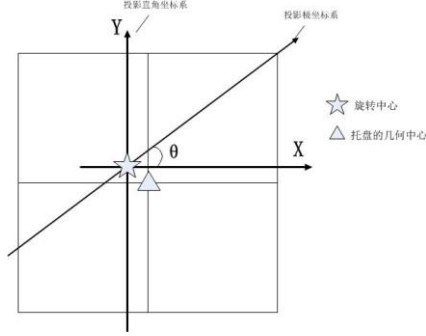


图 11 投影坐标系示意图

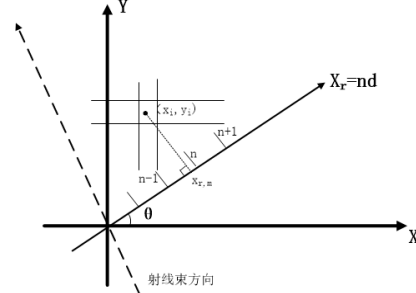


图 12 投影坐标系示意图

(2) 坐标系间转换

设 (x, y) 为投影直角坐标系上任意一点，该点和原点连线与 x 轴夹角为 θ ，线段长度为 ρ ，在投影极坐标系中表示为 (r, θ) ，则

$$r = x * \cos\theta + y * \sin\theta \quad (12)$$

由于滤波过程中的得到的值均为离散值，且投影直角坐标系和投影旋转坐标系的值均为离散值，所以在坐标系间进行转换时需要用插值算法去求近似解，此处我们选用临近插值和线性插值求解得到图像矩阵。

(3) 将矩阵转化为灰度图并显示

由于 MATLAB 显示的灰度值的范围为 $[0, 1]$ ，因此将原矩阵除以 $\frac{\pi}{360}$ ，使得矩阵内元素的取值范围均在 $[0, 1]$ 内；且有雷登反变换之后的得到的矩阵为规模为 362×362 ，因此，要利用双线性插值将矩阵缩放为规模为 256×256 的矩阵。

5.3 问题一模型的求解

(1) 探测器之间的距离

小圆直径为 8mm，则不等关系式为：

$$\frac{8}{29} < d_0 < \frac{8}{28}$$

保留 4 位小数：

$$0.2758 < d_0 < 0.2857$$

在误差允许的情况下，我们用最大值和最小值的平均值作为 d_0 的值，即取

$$d_0 = 0.2808$$

此时，最大误差不超过 1.74%，是可以接受的。因此，本文在接下来的模型中取探测器之间的距离为 0.2808 mm。

(2) 确定旋转中心的位置

由于方程（2）中包含 3 个未知参数，因此我们从附件二数据的图像上选取 3 个特征点，带入方程求解参数。3 个特征点分别为：A. 投影方向沿椭圆长轴的方向，B. 投影方向沿椭圆短轴的方向，C. 小圆轨迹最低点。3 个点位置示意图如下：

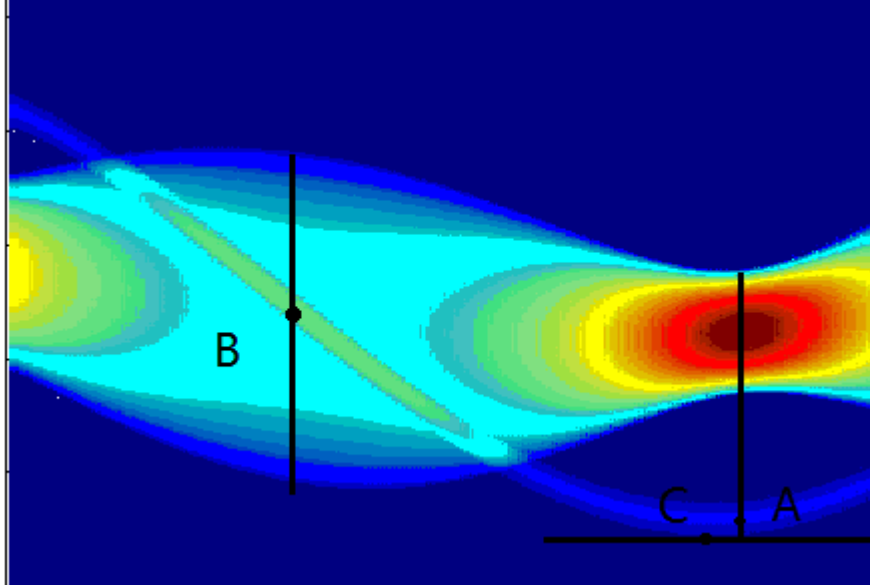


图 13 特征点选取示意图

1) A 点坐标

利用 MATLAB 寻找全局最大值，即为沿椭圆长轴的投影方向，在此方向上寻找小圆的极值点，据此找到满足条件的 1 个点：（151, 60）。即 $y_A = 60$ 。此时，旋转角度 $\theta_A = \pi$ 。

2) B 点坐标

利用 MATLAB 寻找上边界与下边界的最大差值，即为沿椭圆短轴的投影方向，由于探测器之间的间隙产生的误差，找到了满足条件的 8 个点：

表 1 上、下边界的最大差值点

x	58	59	60	61	62	63	64	65
y	245	242	238	235	232	228	225	222

将 y 值取平均作为 B 点的 y 值，即 $y_B = 233.5$ 。此时，旋转角度 $\theta_B = \frac{\pi}{2}$ 。

3) C 点坐标

在附件二数据中可以找到 7 个最低点：

表 2 最低点

x	142	143	144	145	146	147	148
y	59	59	59	59	59	59	59

则 $y_C = 59$ 。轨迹上最低点即为方程变化率最小的点，即此时 $\theta_C + \theta = \pi$ 。

将三个点的信息带入方程（2）中，得到如下方程组：

$$\begin{cases} 59 = -R + C \\ 60 = R \cos(\pi + \theta_0) + 59 + R \\ 233.5 = R \cos\left(\frac{\pi}{2} + \theta_0\right) + 59 + R \end{cases} \quad (13)$$

利用 MATLAB 编程解得：

$$\begin{cases} R = 190.2644 \\ C = 249.2644 \\ \theta_0 = 5.8173^\circ \end{cases}$$

即原方程为：

$$y = 190.2644\cos(\theta + 5.8173) + 249.2644 \quad (14)$$

(3) 确定 180 个投影方向

1) 提取小圆圆心位置

分离段小圆圆心位置的提取结果如图 14 所示；

利用 prewitt 算子进行边缘提取的结果如图 13 所示；

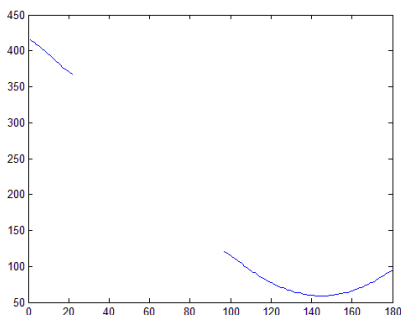


图 14 分离段提取圆心位置图像

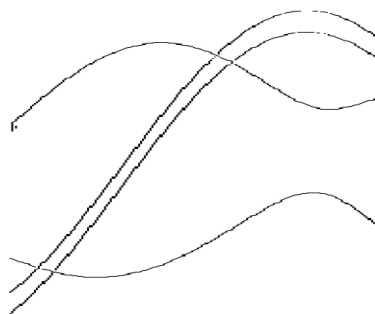


图 13 prewitt 算子提取边界结果

利用 MATLAB 编程从边缘点集提取出所需的小圆的边界，每列求平均值，得到的重合段小圆圆心位置如图 14 所示。

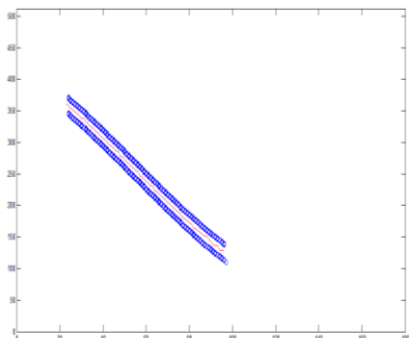


图 14 重合段圆心位置提取图像

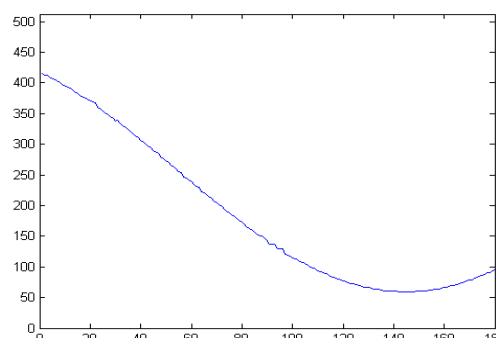


图 15 完整小圆圆心位置坐标

将两段提取出的位置整合，得到完整的 180 个小圆圆心的位置图像，图 15 即为求得的图像。小圆圆心具体位置坐标见支撑材料“小圆圆心 Y 坐标”。

2) 求解 180 个投影方向

将每一个小圆圆心的 y 坐标带入方程 (3)，解出对应的 θ 值，即投影方向。解出的方向范围为： $[22.9792^\circ, 210.0095^\circ]$ 。具体各个方向见支撑材料“问题一角度”。

5.3.4 参数修正

(1) 进一步确定旋转角度的范围

利用方程 (3) 得到的角度范围大致为 187° ，为了得到重建算法需要的精确的 180° 的范围，我们把已经求得的标定参数带入问题二模型，将旋转角度参数划分为： $[23^\circ, 202^\circ]$ 、 $[24^\circ, 203^\circ]$ 、 $[25^\circ, 204^\circ]$ 、 $[26^\circ, 205^\circ]$ 、 $[27^\circ, 206^\circ]$ 、

[28°,207°]、[29°,208°]、[30°,209°]、[31°,210°] 9 个旋转角度范围，分别将附件二的数据进行重建，得到的图像如图 16 所示：

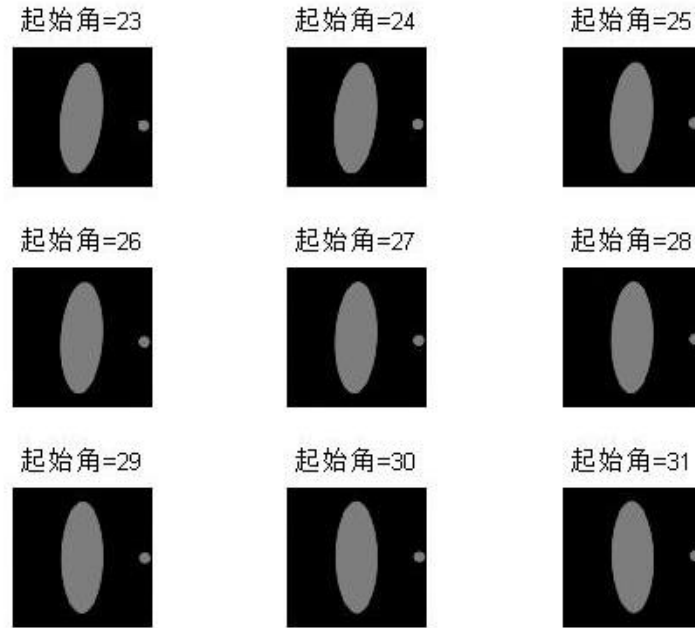


图 16 起始角对比图

观察以上 9 幅图，我们直观地选取 28° 作为起始角，即旋转角度范围为 [28°,207°]。

在重建图像中提取椭圆和小圆的中心点，相应的 MATLAB 程序见附录，结果如图 17：

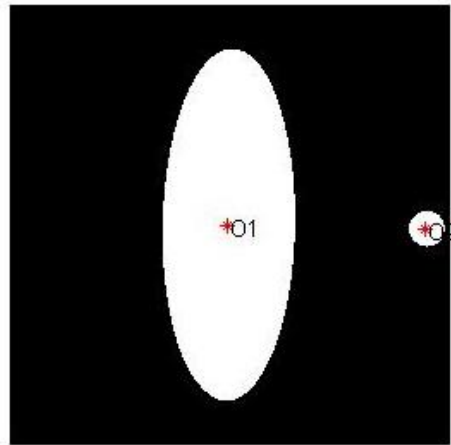


图 17 提取中心点

两中心点在 MATLAB 图像上的坐标分别为： $O_1(184.3208,184.8958)$ ， $O_2(347.1070,189.4517)$ 。发现两点纵坐标并不相等，即起始角度仍有微小偏差，偏差为：

$$\theta' = \arctan\left(\frac{\Delta y}{\Delta x}\right) = 1.6031^\circ$$

即，最终修正之后的旋转角度范围为：

$$\theta \in [29.6031^\circ, 208.6031^\circ]$$

(2) 修正旋转中心坐标

将上述最终的修正角度输入问题二模型，再次将附件二重建，提取椭圆中心点坐标，具体过程与之前叙述相同，输出的结果为： $O_1(184.9501,183.9675)$ 。发现椭圆中心点并不在重建 CT 图像的中心点（181,181）。为了使重建图像的中心点与托盘的几何中心重合，将这个坐标偏差修正到旋转坐标上，则最终修正之后的旋转中心在绝对坐标系中坐标为：

$$\begin{aligned} x &= -8.1511 - d_0 \cdot (184.9501 - 181) = -9.2603(mm) \\ y &= 5.4151 + d_0 \cdot (183.9675 - 181) = 6.2484(mm) \end{aligned}$$

5.4 问题二模型求解

在滤波反投影解算过程中，我们以 MATLAB 库函数 `iradon` 和 `radon` 为基础，根据需要对库函数做了相应的改动，具体的改动在支撑材料 `iradon_reveise` 函数中已经标出。在求解时适时对函数的各个参数进行了修正，使得结果更为精确。

5.4.1 输出重建图像

利用 5.2 中建立的修正滤波反投影重建算法，将附件 3 的数据重建，得到的重建图像如图 18 所示：



图 18 附件 3 滤波反投影图像

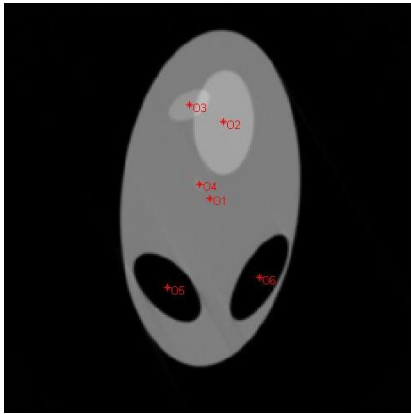


图 19 各椭圆中心点示意图

则附件 3 介质的几何形状如图所示，可以看作一个大椭圆的下部抠掉了两个椭圆，上部还有 3 个小椭圆部分与主体椭圆是不同的介质。

5.4.2 介质在正方形托盘中位置的确定

(1) 提取各个椭圆中心点

利用中心点提取算法提取各个椭圆的中心点如图 19 所示：

在以托盘几何中心为原点的绝对坐标系下各个中心点坐标值为：

表 3 各椭圆中心点坐标值

椭圆编号	01	02	03	04	05	06
x 坐标	1.7791	10.12	-11.0355	-4.7931	-24.8386	32.3053
y 坐标	5.9773	53.4523	63.929	14.9655	-49.2769	-43.275

(2) 确定比例尺

为更加准确表达位置信息，将以像素为单位的相对坐标值转化为以毫米单位的绝对坐标值。搜索到椭圆和小圆的中心点坐标为：

$O_1(128.1354, 128.1375)$, $O_2(243.2566, 128.1375)$, 由此可以得到相邻两个像素点的距离为 0.3909mm。

表 4 各椭圆中心点坐标值

椭圆编号	01	02	03	04	05	06
x (mm)	0.695437	3.955831	-4.31369	-1.87359	-9.70922	12.6279
y (mm)	2.336481	20.8941	24.98936	5.8499	-19.262	-16.9159

5.4.2 吸收率的确定

本题目中的“吸收率”是以附件一中的数据为基准，附件一的介质是均匀介质，因此附件一的数据全部为 0 或 1。在问题一的求解中，我们对附件二进行重建，得到的重建图像中椭圆部分的灰度值在 0.49 上下微小浮动。

为了寻求重建图像的灰度值与附件一的吸收率基准值的关系，我们对附件二原始数据作乘 2 处理，得到新的数据再进行重建，输出图像的椭圆部分的灰度值变为在 0.98 浮动，可以认为实现了灰度值与吸收率的统一。因为重建算法中间过程十分复杂，在研究吸收率时，我们可以把中间过程看成一个黑箱系统，只考虑输入与输出，那么，可以推断出原始数据与吸收率的关系：



图 20 黑箱系统流程图

按照这个规律，我们把附件 3 原始数据作二倍增益处理，输入到重建算法中，得到了附件 3 介质的吸收率，具体值见支撑材料“problem2.xls”，在此我们仅给出题目要求的 10 个点吸收率。

表 5 附件 3 中 10 个特殊点的吸收率

编号	1	2	3	4	5	6	7	8	9	10
吸收率	0	0.9771	0	1.1749	1.0435	1.4231	1.2772	0	0	0

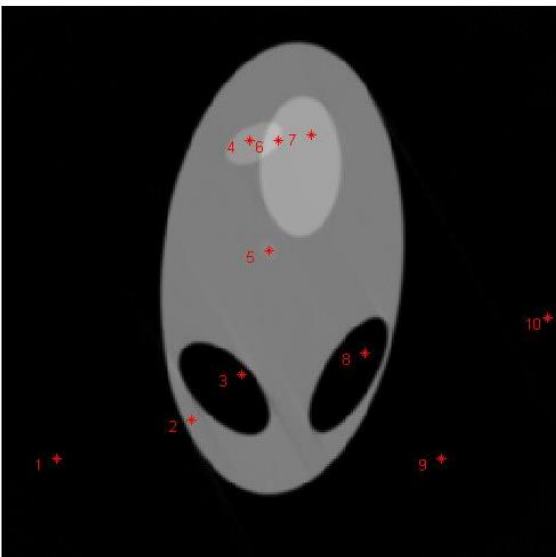


图 21 附件 5 中 10 个特殊点的分布

5.5 问题三的求解

5.5.1 附件 5 的重建图像

利用问题二模型对附件 5 进行重建，得到的重建图像如图 22 所示：

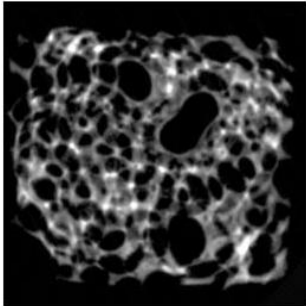


图 22 附件 5 滤波反投影图像

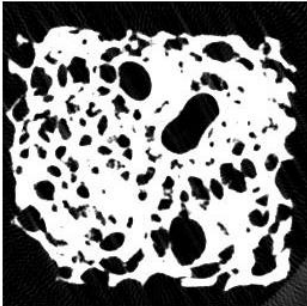


图 23 二值处理后的位置示意图

5.5.2 介质在正方形托盘中位置的确定

由于重建后的图像并不规则，因此，我们将重建后的图像进行二值化处理，如图 23 所示，有物体的点置为 1，空白处置为 0，将该图的像素存储为 256×256 的矩阵，见支撑材料“附件 5 二值化后位置矩阵”。

5.5.3 吸收率的确定

10 个特殊点的位置如图 24。

表 6 附件 5 中 10 个特殊点的吸收率

编号	1	2	3	4	5	6	7	8	9	10
吸收率	0.0452	2.6751	6.8431	0.0140	0.3677	3.1450	6.0678	0	7.3391	0.0437

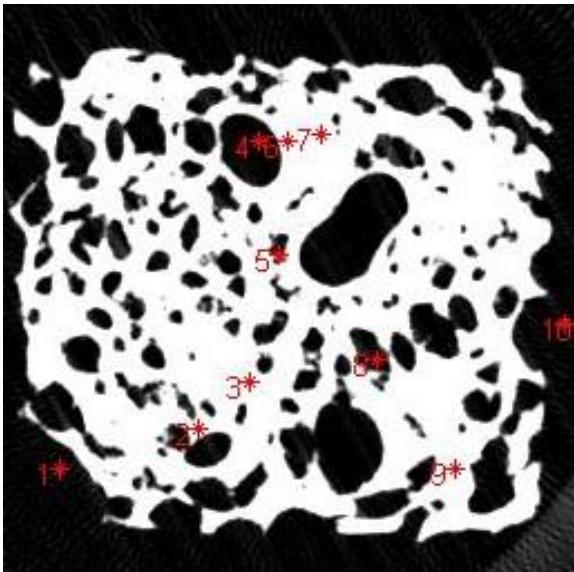


图 24 附件 5 中 10 个特殊点的分布

全部点的吸收率见支撑材料“problem3.xls”。

5.6 问题四模型的建立与求解

(1) 问题一标定参数的误差分析

问题一的核心参数是旋转中心的位置，方程（2）中的 r 、 θ 两个参数反应了旋转中心的位置。问题一的求解模型为：

$$\begin{cases} y_1 = r * \cos(\theta_1 + \theta_0) + C \\ y_2 = r * \cos(\theta_2 + \theta_0) + C \\ y_3 = r * \cos(\theta_3 + \theta_0) + C \end{cases} \quad (15)$$

即，选取 3 个特殊点，将特殊点的值带入方程，3 个方程 3 个参数，可以求得参数的值。

但由于接收信息是离散的，在特殊点的选取过程中采用了取均值处理，这不可避免地会产生误差。另一方面，受椭圆模板自身特性的限制，只能选取到 3 个特殊点，无法取更多点带入模型，因此无法验证模型的准确性。

(2) 新标定模板的设计

针对以上的误差分析，在保留小圆模板的基础上，我们分别设计了正方形和正六边形两种模板来代替原先的椭圆形模板。

1) 正方形模板

正方形标定模板的尺寸、位置信息如图 25：

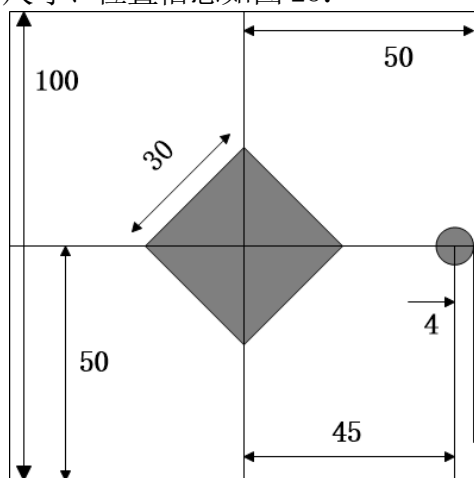


图 25 正方形模板

在旋转一周的情况下，会找到如下四种特殊的照射方向： θ 分别等于

$$\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi。$$

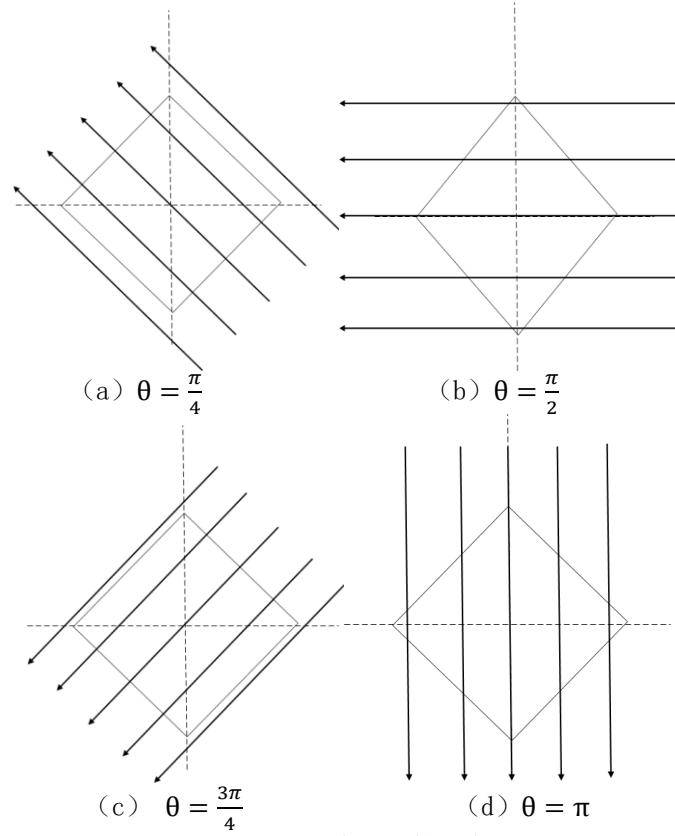


图 26 4 种特殊照射方向

即可以找到 4 个特殊点，代入方程 (2) 得到如下方程组：

$$\begin{cases} y_1 = r * \cos\left(\frac{\pi}{4} + \theta_0\right) + C \\ y_2 = r * \cos\left(\frac{\pi}{2} + \theta_0\right) + C \\ y_3 = r * \cos\left(\frac{3\pi}{4} + \theta_0\right) + C \\ y_4 = r * \cos(\pi + \theta_0) + C \end{cases} \quad (16)$$

2) 正六边形模板

正六边形模板的尺寸和位置信息如图 27：

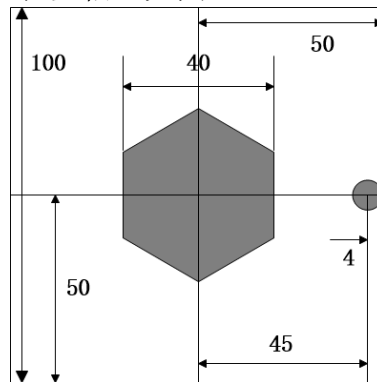


图 27 正六边形模板

在旋转一周的情况下，会找到如下 6 种特殊的方向： θ 分别等于

$$\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}。$$

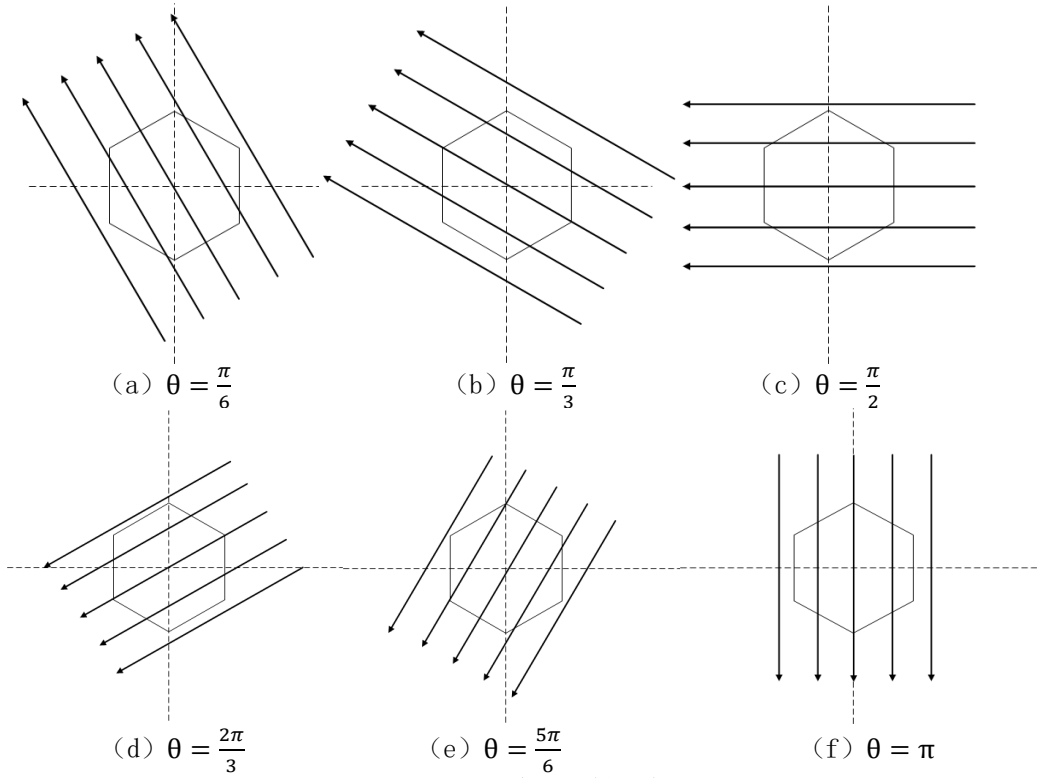


图 28 6 种特殊照射方向

即，可以找到 6 个特殊点，带入方程（2）得到如下方程组：

$$\begin{cases} y_1 = r * \cos\left(\frac{\pi}{6} + \theta_0\right) + C \\ y_2 = r * \cos\left(\frac{\pi}{3} + \theta_0\right) + C \\ y_3 = r * \cos\left(\frac{\pi}{2} + \theta_0\right) + C \\ y_4 = r * \cos\left(\frac{2\pi}{3} + \theta_0\right) + C \\ y_5 = r * \cos(\pi + \theta_0) + C \\ y_6 = r * \cos\left(\frac{4\pi}{3} + \theta_0\right) + C \end{cases} \quad (17)$$

（3）求解方法

以上两个方程组都是方程个数大于未知参数，属于超定方程组。由于方程组是非线性的，因此不能用传统的超定方程组的最小二乘解法求解，但可以采用智能搜索算法去寻求参数的最优解，如遗传算法、蚁群算法和狼群算法等。

（4）新模板的误差分析

按照这个思路，理论上还可以选取正八边形、正十二边形等等作为标定模板，得到的方程个数会依此增加，方程个数越多，智能算法寻优得到的参数值就会越准确。

但另一方面，由于 CT 系统的探测器是离散的，选取的正多边形边数越多，实验得到的原始数据误差就会增大，由实验数据寻找特殊点时的误差也会相应增大。因此，并不是边数越多、方程越多，得到的解就会越精确。

综合考虑以上两个方面，我们主观上折中选取了正方形和正六边形作为新的标定模板，但想要定量地研究其标定精度，还需要真实的实验数据去检验。

六 模型的检验

对问题一中最后修正的旋转中心进行检验：

当 $\theta = \pi$ 时，小圆与探测器平面在探测器坐标系中的的相对运动态势如下所示：

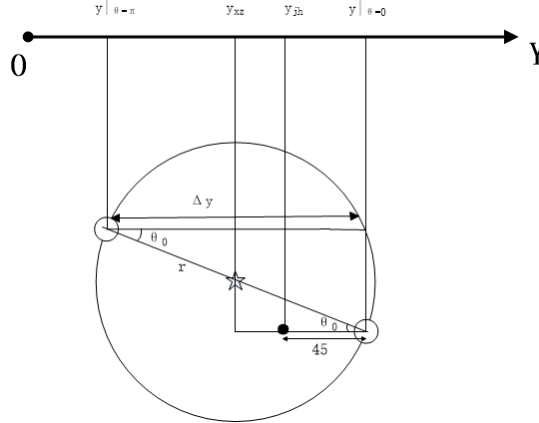


图 29 $\theta = \pi$ 时小圆与探测器平面的相对运动态势
由平面几何关系得：

$$y_{xz} = y|_{\theta=\pi} + \frac{(45 + 9.2603)}{d_0}$$

$$y_{jh} = y_{xz} + r \cos \theta_0 - \frac{45}{d_0}$$

将问题一求得的标定参数 θ_0 、 r 带入以上方程解得：

$$y_{xz} \approx 253$$

$$y_{jh} \approx 286$$

即在初始时刻旋转中心对应第 253 个探测器，几何中心对应第 286 个探测器，而探测器平面的中点应介于第 255 和 256 个探测器之间，则可以算出旋转中心相对于探测器平面中点的误差为 $0.702mm$ ，几何中心相对于探测器平面中点的误差为 $8.5644mm$ 。

分析误差发现修正之后的旋转中心的投影几乎与探测器平面的中心重合，这个结论支持了前文模型的结果。另一方面同时说明了探测器平面在安装时就存在误差，并没有对准托盘的几何中心，在安装时中出现几毫米的安装误差是十分正常合理的，也是符合实际情况的。

六 模型评价与改进

6.1 模型的评价

6.1.1 模型的优点

- 1、通过数学关系推导，模型精确度较高；
- 2、模型假设遵循客观规律，引入误差较少；
- 3、通过精确函数关系计算过程中所需数据，比直接从图中取点精确度高；
- 4、问题二、问题三中对雷登反变换函数进行初始参数修改，将问题一中求解的标定参数在重建图像之前进行修正，同时选用多种滤波函数，使得重建图像目视效果较好。

6.1.2 模型的缺点

- 1、问题一中在小圆与椭圆交界处分界线不清晰，求取小圆圆心误差较大；
- 2、问题二、问题三中推导吸收率和灰度间关系时取点较少，增大了误差。

6.2 模型的改进

- 1、针对缺点一，可通过运用加窗滤波、边界推算、边缘化后选取、滑动窗口选取等多种处理方法分别求交界处圆心，并将结果加权平均，以求得最佳圆心位置。
- 2、可对重建后的灰度值进行离散点求和，再求取其平均值，以获得更精确的函数关系。

七 模型推广

对于第一代 CT 应用平行射线有采样速度慢、采样点多、射线利用率低等缺点，出现了锥形、扇形束射线的投影重建方法。对于扇形束射线的投影重建，分为两种，一种是直接重建，该方法是将投影数据选取适当的权值进行加权平均后得到类似于平行射线的算法；另一种是对扇形数重排后重建，该方法对所有的投影束进行重新组合，形成不同角度下的平行射线投影数据。无论是上述哪种方法，最终都要用到平行射线投影重建的模型。并且对于等间距的接收器排列或等角的接收器排列，都可通过一定的计算方法近似转化为平行射线投影。因此，本文提出的模型同样可用于扇形束射线的投影重建。

八 参考文献

- [1]毛小渊. 二维 CT 图像重建算法研究[D].南昌航空大学,2016.
- [2]孟凡勇,李忠传,杨民,李静海. 基于投影原始数据的 CT 旋转中心精确确定方法[A]. 中国体视学学会.第十三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:,2013:6.
- [3]A. C. Kak, Malcolm Slaney, "Principles of Computerized Tomographic Imaging", IEEE Press 1988.
- [4]Rafael C. Gonzalez, Richard E.Woods. Digital Image Processing, Third Edition. 电子工业出版社.2011.6
- [5]庄天戈.CT 理论与算法[M].上海:上海交通大学出版社,1992,22-76.
- [6]千导.matlab 实现 fbp 算法.http://blog.csdn.net/one_thousand/article/details/23733603?locationNum=3
- [7]Dsp Tian.matlab 练习程序（radon 变换）.<http://www.cnblogs.com/tiandsp/archive/2013/05/24/3096363.html>
- [8]sailor1113. FFT 的详细解释，相信你看了就明白了。。。.<http://www.ilovematlab.cn/thread-119939-1-1.html>
- [9]飞来疯. MATLAB 中 FFT 的使用方法.http://blog.163.com/fei_lai_feng/blog/static/9289962200971751114547/
- [10]睿吉 jerry. MATLAB 把坐标轴(X Y 轴)移到坐标原点.http://blog.sina.cn/dpool/blog/s/blog_4ac35a650100va35.html
- [11]st5302783.答复同学 Matlab roipoly 函数的用法.<http://www.ilovematlab.cn/thread-111922-1-1.html>

附录

1. 问题一程序：

(1) 显示附件一标定模板二值图像：

```
clear all

clc

%% 表格 1 数据读取

A=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛
\CUMCM2017Problems\A\A 题附件.xls');

imshow(A)
```

(2) 显示附件二正弦图 :

```
clear all

clc

%% 表格 2 数据读取

B=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛
\CUMCM2017Problems\A\A 题附件.xls',2);

X=1:size(B,2);
Y=1:size(B,1);

figure(1)
mesh(X,Y,B)
```

(3) 正弦图进行边缘提取

```
clear all

clc

%% 表格 2 数据读取

B=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛
\CUMCM2017Problems\A\生成数据.xlsx',2);

X=1:size(B,2);
Y=1:size(B,1);

imshow(B);

A=edge(B,'prewitt');

imshow(A);
```

(4) 椭圆、小圆分离时小圆圆心提取 :

```
clear all

clc

clear fz j i

%% 表格 2 数据读取
```

```

B=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛
\CUMCM2017Problems\A\A 题附件.xls',2);

X=1:size(B,2);
Y=1:size(B,1);

%% 计算小圆中心点
for j=450:-1:50
    if B(j,1)>B(j-1,1)&&B(j,1)>B(j+1,1)
        fz(1)=j;
        break;
    end
end
for i=2:70
    for j=fz(i-1)-5:fz(i-1)+5
        if B(j,i)>B(j-1,i)&&B(j,i)>B(j+1,i)
            fz(i)=j;
            break
        elseif j==(fz(i-1)+5)
            fz(i)=fz(i-1)-2;
            break;
        end
    end
end
for j=50:150
    if B(j,180)>B(j-1,180)&&B(j,180)>B(j+1,180)
        fz(180)=j;
        break;
    end
end
end

```



```

for i=179:-1:71
    for j=fz(i+1)-5:fz(i+1)+5
        if B(j,i)>B(j-1,i)&&B(j,i)>B(j+1,i)
            fz(i)=j;
            break
        elseif j==(fz(i+1)+5)
            fz(i)=fz(i+1)+2;
            break;
        end
    end
end
end
plot(X(1:22),fz(1:22))
hold on
plot(X(97:180),fz(97:180))

```

(5) 椭圆、小圆重合时小圆圆心提取 :

```

clear all
clc
%% 表格 2 数据读取

B=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛\CUMCM2017Problems\A\A 题附件.xls',2);

X=1:size(B,2);
Y=1:size(B,1);
A=edge(B,'prewitt');
imshow(A);
C=double(A);

%消除噪点

for i=1:8
    C(358+i,i)=0;

```

```

    C(357+i,i+8)=0;
end
C(373:378,17:22)=0;
C(366:372,9:14)=0;
C(377:380,23:28)=0;
C(93:101,91:100)=0;
C(103:110,102:108)=0;
C(113:136,110:124)=0;
%% 处理连续 1
x=1:180;
ysx=-170*cos(pi*x/180-5*pi/6)+290;
yxx=-170*cos(pi*x/180-5*pi/6)+180;
js=1;
for i=1:180
    n=0;
    fz=0;
    for j=floor(yxx(i)):ceil(ysx(i))
        if C(j,i)==1
            x0(js)=i;
            y0(js)=j;
            js=js+1;
            n=n+1;
        end
    end
    end
    N(i)=n;
    ypj(i)=sum(y0(js-n+fz:js-1+fz))/n;
    ypj(1)=416;
end
end

```

```

for i=1:length(x0)
    if x0(i)>23
        break
    end
end
for j=1:length(x0)
    if x0(i)>96
        break
    end
end
plot(x0(i:579),y0(i:579),'o')
hold on
plot(X(23:96),ypj(23:96),'r-')
axis([0 180 0 512])

```

(6) X 射线角度求取

```

clear all
clc

%% 生成数据读取

y=xlsread('D:\QQPCmgr\Documents\国赛模拟\国赛\CUMCM2017Problems\A\生成数据.xlsx',4);

X=1:180;

%% 求解角度

for i=1:180
    if i>141
        sita(i)=360-acosd((y(i)-249.4933)/190.4933)-5.8734;
    else
        sita(i)=acosd((y(i)-249.4933)/190.4933)-5.8734;
    end
end

```

```
end
```

```
plot(X,sita);
```

(7) 求解 sita0

```
%-----
```

```
fun = @(x) 174.5*cosd(x)-173.5-sind(x); % function
```

```
x0 = 4; % initial point
```

```
x = fzero(fun,x0)
```

2.问题二程序:

```
% code 2.1
% 标题: 问题二、三求解
% 时间: 2017.9.17
clear;close;clc;clf;
% 滤波反投影
P =xlsread('A 题附件.xls',5);
for i=1:9
    theta =23+6.5731:1:23+179+6.5731;%6.5731 %0:1:179;%投影角度+5
+1.5654;-3 +1.5654
    theta_num = length(theta);
    d = 1;
    I = iradon(P*2,theta,'hamming',1);
    I256 = imresize(I,[256,256],'bilinear');%采用双线性插值放缩为
256*256
    subplot (3,3,i);
    imshow(I256);
    title(['起始角=',num2str(22+i)]);
end
xlswrite('3.4 吸收度矩阵.xls',I256)
hold on
title('滤波反投影');
%% 10个吸收点的坐标
xx=[25.66994464      209.8640443
88.34710355      191.9562846
```

```

111.371366    171.4902735
115.2087431    62.76458971
124.1626229    113.9296174
128        62.76458971
143.3495083    60.20633832
167.6528965    161.257268
203.4684158    209.8640443
252.0751921    144.628634];
for i=1:10
    xishou(i)=I256(round(xx(i,2)),round(xx(i,1)));
    plot(round(xx(i,1)),round(xx(i,2)),'r*');
    text(round(xx(i,1))-
10,round(xx(i,2))+3,[num2str(i)],'color',[1 0 0]);
end
xlswrite('3.5xishoulv.xls',xishou);
%% 输出中心标号
position_o=[129.7791    122.0227
    138.12    74.5477
    116.9645    64.071
    123.2069    113.0345
    103.1614    177.2769
    160.3053    171.275    ]
for i = 1 : length(position_o)
    temp=position_o(i,:);
    plot(temp(1), temp(2), 'r*');
    text(temp(1)+2, temp(2)+2,['0',num2str(i)],'color',[1,0,0]);
end
%% 图像二值化处理
II=zeros(size(I256));
I1 = im2bw(I256,0.2);%将图像进行二值化处理，在（0,1）范围内，为下
界
I2 = ~im2bw(I256,1);%将图像进行二值化处理，在（0,1）范围内，为上
界
II=I1&(I2);%两图像相与，取得合成值
figure
imshow(II);
III=II;%有重影时使用的临时变量
II=III|II;
%% 多边形选区的选取（只有在求 01 时才用）
nn=1 ; %要选取的区间的个数
gains=zeros(size(II));
for i=1:nn
    figure

```

```

h=imshow(II);
title(['第', num2str(i), '次多边形区域选择'])
hold on;
x=[];y=[];
[x,y,c]=ginput(1);
m(1)=x;
n(1)=y;
plot(x,y,'r');
k=2;
while(c==1)
    [x1,y1,c1]=ginput(1);
    if c1==1
        m(k)=x1;
        n(k)=y1;
        plot(x,y,'r');
        line([m(k-1) m(k)], [n(k-1) n(k)]);
        k=k+1;
        c=c1;
    else
        break
    end
end
line([m(k-1) m(1)], [n(k-1) n(1)]);
BW = roipoly(II,m,n);
gains=gains|BW;%图像累加
pause(0.1);
disp('请勾选下一个')
hold off
end
gains=gains;
imshow(gains)
II=II|gains;%gains %将选区加入到原图中
% 图像属性解算——中心点解算
figure
imshow(II);
hold on
L = bwlabel(II);%将相连的图形进行编号处理，所以分为不同的像素块
stats = regionprops(L, 'Centroid');
for i = 1 : length(stats)
    temp = stats(i).Centroid
    plot(temp(1), temp(2), 'r*');
    text(temp(1)+2, temp(2)+2, ['0', num2str(i)]);
end
end

```

```

hold off;
%% 偏差修正
O1=stats(1).Centroid %椭圆的中心点位置
O2=stats(2).Centroid; %小圆的中心点位置
delta_x=O2(1)-O1(1); %x 方向的偏差
delta_y=O2(2)-O1(2); %y 方向的偏差
tana=delta_y/delta_x; %角度偏差
a=atand(tana);

```