CT 系统参数标定及成像

摘要

本文针对 CT 系统参数标定的问题，结合 Radon 变换的相关知识，根据附件 1、2 的信息完成了本题 CT 系统参数的标定，以此确定了附件 3、5 对应的未知介质信息，并分析了本标定模版的精确性和稳定性，提出了一种更优异的标定模版。

对于问题一，首先通过拟合将椭圆与小圆的接收信息分开，将小圆的各组接收信息拟合求得直径与接收器间距的对应关系，经过 180 组的平均后，求出探测器单元之间距离。然后分别将 180 组椭圆和小圆的接收信息拟合得到函数最高点，即为此角度时两中心的位置（以接收器单元为单位），作差后与换算后的两中心最大距离（以接收器单元为单位）作比，通过反三角函数得到每一组的具体角度。最后通过求出的具体角度，拟合出两个中心的三角函数，其振幅即为对应的中心到旋转中心的距离，再利用旋转方向的信息，即可确定旋转中心。

对于问题二，首先我们根据 iRadon 函数重建的图像，求解出由于外部因素产生的仿射变换。为了求得更精确的 256*256 的重建矩阵，我们放弃使用难以修改的内置函数 iRadon，查阅资料发现滤波反投影算法更加灵活，我们修改了其中的代码使得 FBP 重建 256*256 的图像只需做平移和旋转变换，无需放缩，而且在我们修改后的函数中，仿射变换定义在矩阵重建之前，精度可以保证。针对问题二，由于重建后图像的特殊性，我们假定介质是分区域均匀的介质，因此定义了一个函数用于过滤由 Radon、iRadon 变换产生的噪音。如此我们就得到了与图像几何关系相对吻合的吸收率。

对于问题三，与问题二类似，我们使用调整后的滤波反投影算法得出 256*256 的吸收率矩阵，再用插值的方法求出对应点的吸收率，使其有较高的精确度。

对于问题四，首先对第一问精度的分析，我们主要从求探测器单元间距切入，利用已求得的数据选取另一种新算法重新计算，对比二者之间的差距来评价精度。而之后的角度以及旋转中心位置的精度误差也主要由探测器单元间距产生。并且我们认为人工摆放标定模型也会对获得的参数造成很大影响。

其次是稳定性，我们主要从角度的选取入手，考虑到减少旋转角度的范围对参数标定的影响来衡量稳定性。

基于以上几点分析，我们在之前的基础上拟定了一个新的模型，该模型不再需要特殊的摆放方法，减少了标定的操作难度，其次，我们优化了计算探测器单元间距的精度，也提高了稳定性。

关键词：反 Radon 变换　正弦拟合　滤波反投影重建法　matlab

1. 问题重述

1.1 问题的背景

CT 是一种利用样品对射线能量吸收特性，获取样品内部结构信息的方法，在现代医学和工业检测中有非常广的应用。但在 CT 系统安装时往往存在误差，影响成像质量，故需要借助模版标定 CT 系统的参数，并以此确定未知样品的内部结构。

1.2 相关信息

本题给出了用于标定系统的模版信息，CT 系统对应于该模版的接收信息，以及问题二与问题三中未知介质对应的接收信息。另外本体亦给出了问题二、三中需要求解未知介质的 10 个位置坐标。

1.3 需要解决的问题

1）建立模版几何信息与接收信息的数学模型，分析接收信息的变化规律和其与模版集合信息的关联，确定 CT 系统旋转中心的位置，探测器单元之间的距离以及 CT 系统使用的 X 射线的 180 个方向。

2）利用 1）中的标定参数和未知介质的接收信息，确定未知介质的位置、几何形状、吸收率以及 10 个具体位置的吸收率。

3）与 2）类似，利用 1）中的标定参数和未知介质的接收信息，确定未知介质在 10 个具体位置的吸收率。

4）分析 1）中用于参数标定的模版，分析其精度和稳定性。并在此基础上提出新的模版以改进精度和稳定性。

2 模型假设

假设 1：实验收集的数据能客观准确反映实际情况

假设 2：X 射线始终能覆盖托盘上的物体

假设 3：光束波动不明显，忽略光的衍射现象

假设 4：不存在由于物体表面反射而使 X 射线能量衰减的情况

假设 5：问题二中物体密度除明显的边界外，其余部分均匀

3 符号说明

符号	符号说明
p_θ	X 射线角度为θ时介质的吸收系数
$\rho(l)$	积分路径上每点的密度
num	X 射线角度序数
d	小圆直径（探测器单元单位）
R	探测器单元间距

4 问题的分析

4.1 问题一的分析

针对问题一首先为了求解探测器单元之间的距离，应先确定两均匀固体介质

的中心在 CT 系统接收信息中的轨迹。由于两均匀固体介质的接收信息有重叠部分，所以拟通过椭圆介质未重叠部分的吸收率拟合出重叠部分椭圆介质的吸收率。可得到每组接收信息中椭圆介质与圆介质的中心位置和圆介质直径与探测器单元间距的关系，由后者可得到探测器单元之间的距离。

针对 CT 系统的 180 个方向，由模板中即两中心的最大距离，与 180 组拟合出的两中心垂直距离的关系，利用反三角函数即可确定

针对旋转中心的位置，应先确定两条由 180 个方向的中心确定的拟合曲线，为三角函数。三角函数的振幅即为旋转中心与两介质中心的距离，可得旋转中心在正方形托盘上的位置。

4.2 问题二的分析

为了得到满足要求的 256*256 的重建矩阵，考虑认为，若直接 iRadon 变换得到 362*362 的重建矩阵再通过仿射变换缩减到 256*256 会使一些数据在某些地方堆积而在某些地方稀释，这令人失望。

为了改进方案，我们查阅了现在 CT 机常用的滤波反投影算法（FBP）[1]，由于 iRadon 是 matlab 自带的函数，某些参数无法设置（如 iRadon 后的重建矩阵像素大小），而 FBP 是自编的，可以在其中设置参数，甚至可以将仿射变换直接写在代码中间使得修改后的算法直接能得到第二问那种进行两次操作后才能得到的代码。由于我们直接设置了重建矩阵大小为 256*256，使得从 FBP 重建图像到问题所需的格式只差了一个平移与旋转（无需放缩）。而具体代码将仿射变换写在矩阵重建之前，所以吸收率的精度是可以相信的。

针对第二问，未知介质在托盘中的中的位置与几何信息以及各点的吸收率可以直接通过反 Radon 变换求出，但是由于其特殊性，我们假定该介质应为一个相对均匀的介质，也就是说通过反 Radon 变换得到的图像中的噪音误差应被抹去。具体方法如下：

先进行 RL 滤波的反 Radon 变换，得到 256*256 维矩阵，再根据观察划定各个均匀区域的阈值，然后对每个点进行分类，对每一类点求平均值，作为该类点的统一值，如此就得到了几块平滑的区域，每块的吸收率值都是统一的，这也很好地符合了我们的假设。

而对于所要求的 10 个点的具体吸收率信息，我们通过将 100*100 图像中的点变换到 256*256 的图像中，直接取最近点的方法求出对应点的吸收率，但由于每块区域的值都是确定的，事实上我们只需要判断所要求的点在哪个区域即可。

4.3 问题三的分析

针对第三问，未知介质在托盘中的中的位置与几何信息以及各点的吸收率可以直接通过反 Radon 变换求出，而对于所要求的 10 个点的具体吸收率信息，我们通过将 100*100 图像中的点变换到 256*256 的图像中，再用插值的方法求出对应点的吸收率，使得其有较高的精确度。

4.4 问题四的分析

基于第一问所建立的模型，其精度误差主要体现在探测器单元间距的求解上。所以我们用新的算法的得到的结果作为衡量精确度的标准，从而对精确度进行分析。而对于稳定性，主要从角度考虑，在改小角度范围以及角度的均匀性情况下分析标定参数的稳定性。

5　模型的建立与求解

5.1　问题一

5.1.1 模型的建立

经查资料，我们可知 X 射线通过介质时的吸收系数：

$$p_\theta = \int_d \rho(l)dl \tag{1}$$

其中 p_θ 为 X 射线角度为 θ 时介质的吸收系数，d 积分路径为 X 射线通过的介质路径，$\rho(l)$ 为路径上每点的密度。因此，椭圆和小圆重叠部分的数据可由两部分的吸收系数直接相加。

本文拟用附件 2 中的椭圆部分的数据拟合出每个方向中椭圆中心的位置，这需要我们先去除重叠的数据部分，在此我们选用一个较粗略的函数描述小圆中心轨迹：

$$mid = \left[-200\sin(\frac{\pi}{180}\cdot(num-55)) + 258 \right] \tag{2}$$

经检验，在每列数据中 $[mid-25, mid+25]$ 能完全覆盖小圆的吸收信息。我们将附件 2 中数据去除上述可能重叠的部分，再对每列关于公式：

$$y = \sqrt{ax^2+bx+c} \tag{3}$$

进行拟合，得到 $x_0 = -\frac{b}{2a}$ 为椭圆的中心位置，以及利用拟合函数求得上述重叠部分信息中椭圆的部分。由此作差可得小圆对应的 180 个角度的吸收数据。

由于 X 射线照射小圆时得到的两边界距离始终对应小圆直径 8㎜，我们对其每个角度的吸收数据关于公式（1.3）进行拟合，求得对应的两边界距离和函数曲线最高点即小圆中心的坐标。计算 180 组两边界距离的平均，将其与小圆直径对应，即可得到探测器单元之间的距离。

求解 CT 系统 X 射线的 180 个方向，我们选用椭圆和小圆的中心距离 a 和垂直 X 射线方向的两中心距离 b 为重要指标，通过反三角函数求得射线方向，具体过程如下所述。

用探测器单元之间的距离，将两圆中心的实际距离长度换算为以探测器单元为单位的长度，利用拟合求得的 180 组两圆中心位置之差和实际距离长度带入公式

$$\theta=\arcsin(\frac{p}{q}) \tag{4}$$

即可得该方向对应的角度。

利用求得的角度，我们可以用公式

$$y=a\sin(bx+c)+d \tag{5}$$

分别拟合椭圆和小圆的中心轨迹，得到两个三角函数的振幅即为两中心到旋转中心的距离（探测器单元单位），经换算得到两个以 mm 为单位的半径r_1,r_2，以左上角为原点，以右为 x 轴正方向，下为 y 轴正方向，带入公式：

$$\begin{cases}(x-50)^2+(y-50)^2=r_1^2\\(x-95)^2+(y-50)^2=r_2^2\end{cases} \tag{6}$$

分析 CT 系统逆时针的旋转方向，即可舍去一个错误值，得到旋转中心在正方形托盘中的位置。

5.1.2 模型的求解

问题一的求解过程代码文件为 main1.m，自定义函数 unzip.m, filling.m, unzip_circle.m，其具体作用见 README.txt。

5.1.2.1 求解探测器单元间距

首先将 A 题附件的数据导入 matlab，将附件 2 中数据导入数组 AS1，对 180 个方向，去除重叠部分得到椭圆的数据$\begin{bmatrix}A & B\end{bmatrix}$，其中$B$为探测器序数，$A$为该点吸收率，将其按照公式

$$y=\sqrt{ax^2+bx+c} \tag{7}$$

拟合（图 1-1、1-2），得到 180 组系数，存入矩阵 Ellipse。接下来利用 Ellipse 中系数拟合的曲线与有重合的数据作差，得到圆的吸收数据，同理用公式（1.7）拟合数据（图 1-3、1-4），得到 180 组系数，存入矩阵 Circle。由系数可由公式

$$d=|x_1-x_2|=\frac{\sqrt{b^2-4ac}}{|a|} \tag{8}$$

计算得 180 组直径长度（探测器单元单位），取平均后与直径 8mm 相比，得到探测器单元间距为

$$R=0.2768\text{mm} \tag{9}$$

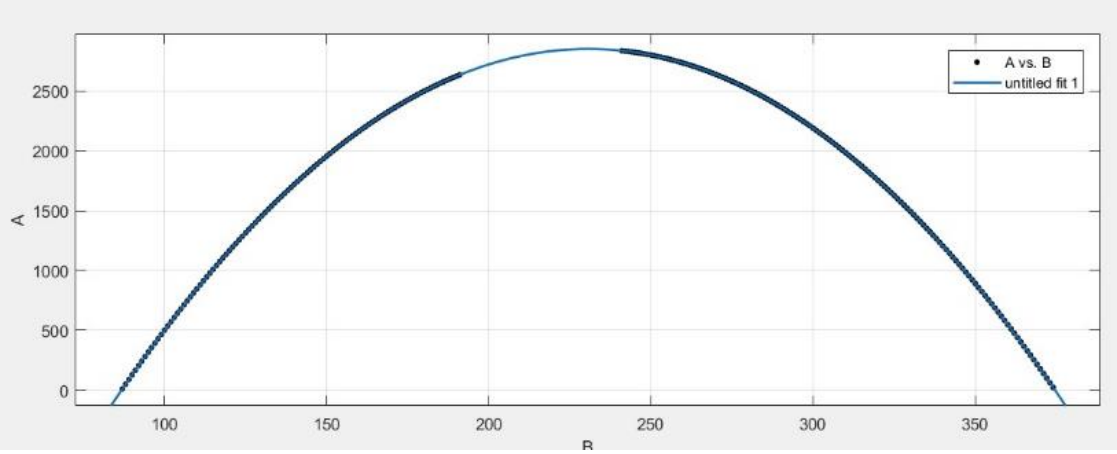

图 1-1 椭圆拟合示例图

Results

Linear model Poly2:

f(x) = p1*x^2 + p2*x + p3

Coefficients (with 95% confidence bounds):

p1 = -0.1376 (-0.1376, -0.1376)

p2 = 63.49 (63.49, 63.49)

p3 = -4471 (-4471, -4471)

Goodness of fit:

SSE: 0.001465

R-square: 1

Adjusted R-square: 1

RMSE: 0.002492

图 1-2 椭圆拟合数据

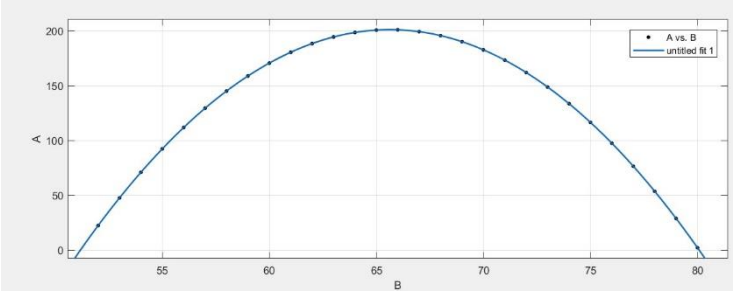

图 1-2 圆拟合示例图

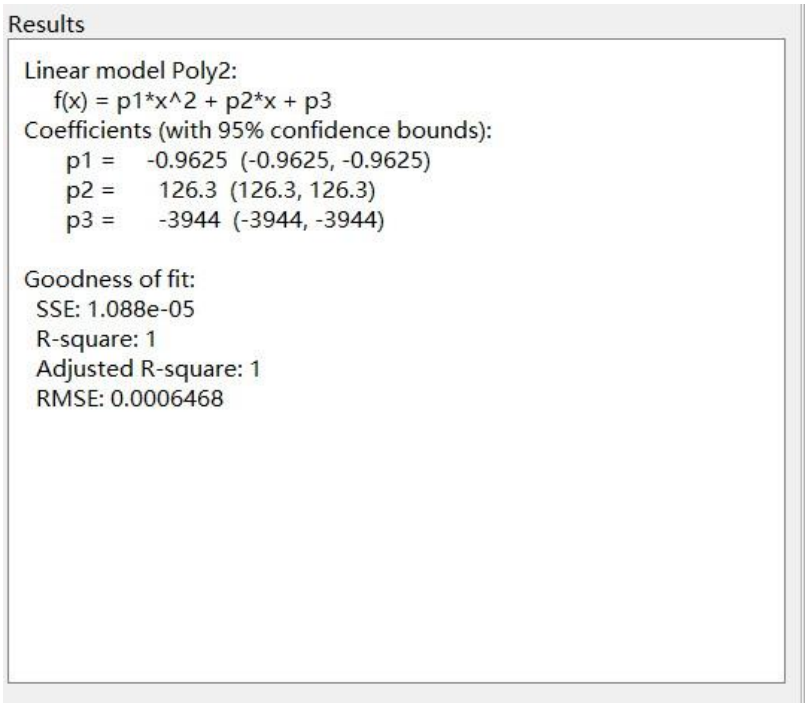

图 1-4 圆拟合数据

5.1.2.2 求解 X 射线的 180 个方向

由拟合系数矩阵 Ellipse 和 Circle 可求得椭圆和小圆的中心位置，分别存入数组 center_Ellipse、center_Circle，并将对应中心位置作差得到数组 center_Distance。由公式（1.4）可得 180 组 X 射线的方向，存入数组 degree。（结果输出在 degree.xls）

5.1.2.3 求解旋转中心

由求得的中心位置数组 center_Ellipse、center_Circle 用公式

$$y = a\sin(bx + c) + d \tag{10}$$

分别拟合（图 1-3、1-4），两个正弦函数的振幅即到旋转中心的距离，椭圆、小圆到旋转中心的距离分别为 $r_1 = 11.1899$，$r_2 = 54.6277$，带入公式（1.6）可得旋转中心坐标为 $\begin{cases} x = 40.7337 \\ y = 43.7271 \end{cases}$（单位：mm），坐标系以正方形托盘左上角为原点，以右为 x 轴正方向，下为 y 轴正方向。

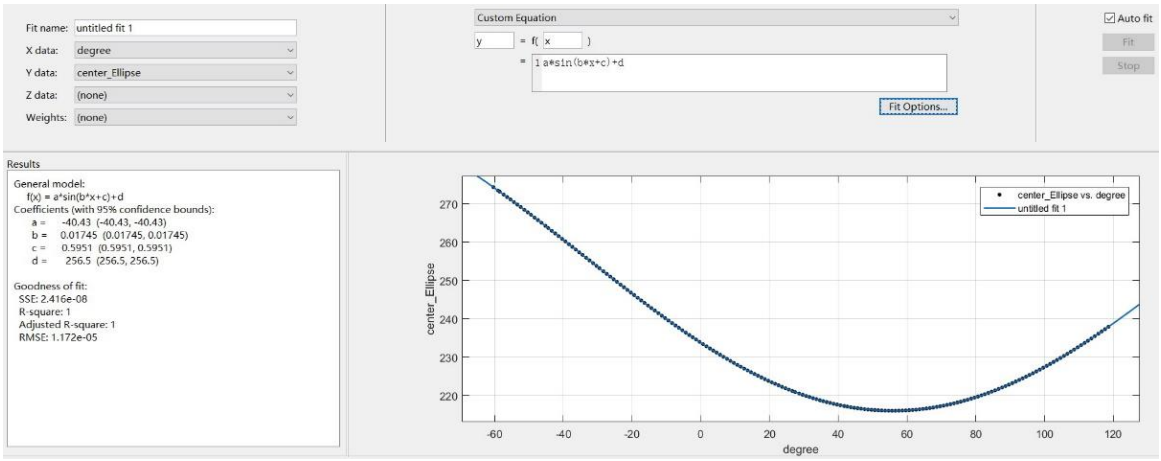

图 1-3 椭圆中心拟合图

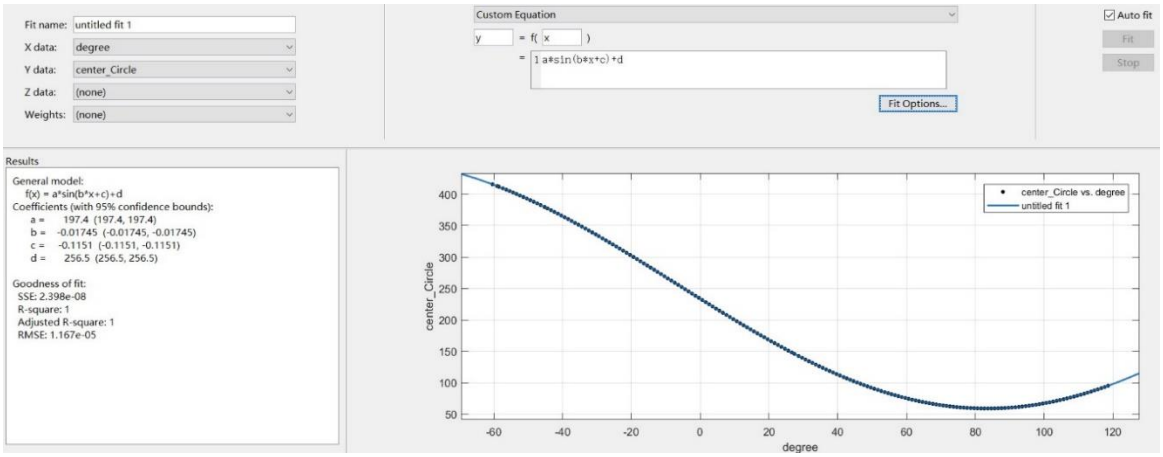

图 1-4 圆中心拟合图

5.1.3 结果分析

本题中，从四个拟合图像中可看出，相关系数都为 1，标准差分别为 0.002492、0.0006484、1.172e-05、1.167e-05，拟合数据的浮动范围很小，说明拟合过程的误差极小，基本还原了参数。但在求解角度时，与 arcsin 函数相关的系数一个来自理论数值，一个来自实际测得值，如果原数据存在误差，可能存在放大误差的可能性。

5.2 问题二

问题二的求解过程代码文件为 main23.m，自定义函数 average.m，filternumber.m、RLfilter.m、RL_FBP.m，其具体作用见 README.txt

5.2.1 模型的建立（FBP 理论）

由 iRadon 变换公式：

$$f(x,y)=\int_0^{\pi} g(t,\theta)d\theta \tag{11}$$

其中

$$g(t,\theta)=\int_{-\infty}^{+\infty}|\omega|P(\omega,\theta)e^{j2\pi\omega t}d\omega=h(t)*p(t,\theta) \tag{12}$$

为滤波修正后的投影函数。式中，$h(t)$是传递函数$H(\omega)=|\omega|$的傅里叶逆变换，$p(t,\theta)$是传递函数$P(\omega,\theta)$的傅里叶逆变换。

卷积的离散化公式为

$$g(n,m)=p(n,m)*h(n)=\sum_{l=-N_l}^{N_l}p(n-l,m)h(l) \tag{13}$$

同时 iRadon 的离散化为

$$f(r,\phi)=\frac{\pi}{M}\sum_{m=0}^{M-1}\tilde{p}(t_m,\theta_m) \tag{14}$$

写成递归形式

$$f_m(i,j)=f_{m-1}(i,j)+\tilde{p}[\tilde{t}_m(i,j),m\Delta],\ m=1,2,\cdots,M \tag{15}$$

以方便编程实现

为了计算上式中的累加项，采用线性插值法。为了计算方便，考虑一个 N*N 栅栏，而 （N/2，N/2）为其坐标原点，区别于像素原点。

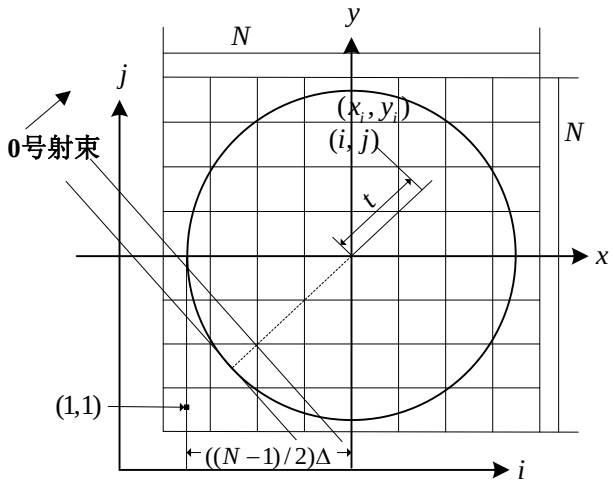

图 2-1 坐标 *x-y*，*i-j*,以及$t,\tilde{t}$的关系说明图

我们的目的是根据任意像素坐标（*i*，*j*）求得在某个视角上的新的坐标 t，而这可由简单的计算得到：

$$\begin{aligned}t&=x_i\cos\theta+y_j\sin\theta=(i-\frac{N}{2})\cos\theta+(j-\frac{N}{2})\sin\theta\\&=(i-1)\cos\theta+(j-1)\sin\theta-\frac{N}{2}(\cos\theta+\sin\theta)\end{aligned} \tag{16}$$

$$
\begin{aligned}
\tilde{t} &= t + \frac{N}{2} \\
&= (i-1)\cos\theta + (j-1)\sin\theta - \frac{N}{2}(\cos\theta + \sin\theta) + \frac{N}{2} \\
&= (i-1)\cos\theta + (j-1)\sin\theta + \frac{N}{2}(1 - \cos\theta - \sin\theta) \\
&= (i-1)\cos\theta + (j-1)\sin\theta + C_\theta(\text{常数}) \\
&= \text{整数}(n_0)\text{+小数}(\delta)
\end{aligned}
\tag{17}
$$

t 的整数部分对应视角 theta 上数据编号，而小数部分可用来将那数据与相邻数据进行插值，由此计算过程已经十分明朗。

相应的代码实现：先编写 RL 滤波函数，对原样本进行滤波，然后编写 RL 反投影函数，直接重建反投影矩阵。具体按照离散化 iRadon 变换的叠加原理，先取一个格点，做仿射变换，做其某投影角的直线，计算直线编号，用滤波后的数据进行差值，差值得到的结果放到对应重建矩阵元中。由于此过程将原先的 512 探测器单元数据放到了 256 的矩阵中，为了充分利用数据减少误差，具体代码中，将某条直线编号对应的数值与其相邻的那条直线数据取平均，由此完成滤波反投影。

5.2.2 模型的求解

具体求解问题二时，我们还需进一步标定阈值，以及各个区域所赋的值。先用 main23.m 程序对附件 3 中的矩阵进行 iRadon 变换，从 iRadon 变换后的矩阵中可以大致给出 filternumber.m 程序中的阈值。

然后通过一些范围的修改，在 average.m 程序中遍历 256*256 矩阵中的所有点，一一分类，然后求各个类的吸收率的平均值，作为该类统一的吸收率。

然后再次遍历矩阵，重新一一给各个类赋值，得到的就是一个分成几个区域的矩阵，每个区域有不同的值。然后对于附件中给出的 10 个点进行坐标变换到 256*256 的图像中，由此求出对应点的吸收率。

未知介质在正方形托盘位置、几何形状以及附件中所给 10 个点对应的吸收率如下（图 2-2、表 2-1、表 2-2），对应 256*256 的重建矩阵见 problem2.xls。

图 2-2 问题二中未知介质几何形状

	X 坐标	Y 坐标
最高点	55.0781	6.25
最低点	49.6094	88.6719
最左点	28.9063	52.3438
最右点	73.4375	43.3594

表 2-1 以左上角为原点的位置信息表

X 坐标	Y 坐标	过滤前吸收率	过滤后吸收率
10	18	0.0078	0
34.5	25	0.9720	0.9685
43.5	33	0.0047	0
45	75.5	1.1712	1.1852
48.5	55.5	1.0442	1.0443
50	75.5	1.4519	1.4585
56	76.5	1.2840	1.2768
65.5	37	0.0031	0
79.5	18	-0.0064	0
98.5	43.5	-0.0020	0

表 2-2 问题二中未知介质 10 个位置处的吸收率表

5.2.3 结果分析

本题中，我们希望通过用统一每一类点的方法来减小误差，事实上，我们也给出了不利用阈值，直接用插值的方法求 10 个点吸收率的结果，但是不难看出有几个很接近 0 的点，甚至还有负值，这是不符合常识的，而我们通过分类的方法就极大地减小了误差。

但是不可避免的就是每一个阈值的选取有一定的不可控性，还是有一些边界上的点有着很大的抖动，所以取平均的过程可能会因此产生误差，但是这个误差相对整体来讲是小的。

5.3 问题三

问题三的求解过程代码文件为 main23.m，自定义函数 RLfilter.m、RL_FBP.m，其具体作用见 README.txt。

5.3.1 模型的建立

基本理论同问题二，但是此时由于整体图像呈非几何体性态，我们很难再用取阈值的方法对其进行重构和误差修正了。所以我们就采用了对坐标变换后的点直接就近取插值的方法求解各个点的吸收率。但是由于吸收率不应小于 0，我们直接将所有取负值的点为 0。

5.3.2 模型的求解

同样用 main23.m 程序对附件 5 中的矩阵进行 iRadon 变换，得到 256*256 的矩阵，再将所求点从 100*100 图像中坐标变换至 256*256 矩阵中，利用插值的方法求出对应点的坐标，若该点吸收率小于 0.01，视为由 Radon 变换和 iRadon 变换带来的误差，记为 0。

所得未知介质的信息以及对应 10 个点的吸收率如下。

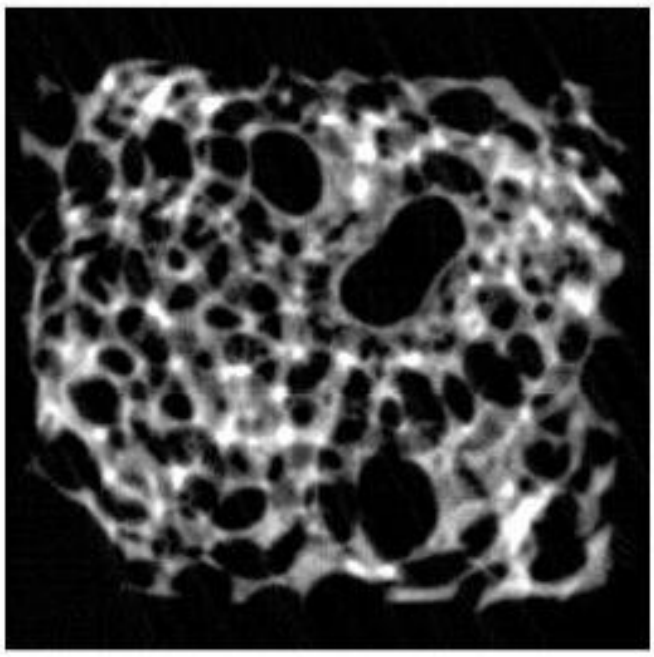

图 2-2 问题三中未知介质几何形状及在托盘上的位置

X 坐标	Y 坐标	过滤前吸收率	过滤后吸收率
10	18	0.1369	0.1369
34.5	25	2.5424	2.5424
43.5	33	6.7467	6.7467
45	75.5	-0.0460	0
48.5	55.5	0.5525	0.5525
50	75.5	2.9533	2.9533
56	76.5	5.5042	5.5042
65.5	37	0.0915	0
79.5	18	6.6309	6.6309
98.5	43.5	0.0707	0

表 3-1 问题三中未知介质 10 个位置处的吸收率表

5.3.3 结果分析

由于本身 iRadon 变换会产生误差，所以此处的吸收率只是大致的一个范围，但是整体趋势以及各个峰谷值之间的对比还是具有很大的参考价值。

5.4 问题四

5.4.1 问题一中模版的分析

5.4.1.1 精度

基于第一问所建立的模型，其精度误差主要体现在探测器间距的求解上。我们考虑再用 main1.m 中的 center_Distance(也就是椭圆和圆圆心在探测屏是上的距离)和 degree（也就是 180 个角度值）拟合出一条正弦曲线如图，那么他的振幅 A=162.6 就是椭圆和圆两圆心的距离（单位为探测器间距）。而实际上两圆心相距 45mm，从而用

$$d=\frac{45}{A} \tag{18}$$

得到探测器间距为 0.276754，与之前所求的 0.276752 差距为

$$\frac{\Delta d}{d}=\frac{0.276754-0.276752}{0.276752}=0.00007\% \tag{19}$$

说明这是很精确的。但是与四舍五入的 0.2768 差距为

$$\frac{\Delta d}{d}=\frac{0.2768-0.276754}{0.2768}=0.02\% \tag{20}$$

相对会大一些，但是也并无大碍。而

$$x=\frac{A_1^2d^2-A_2^2d^2+95^2-50^2}{90} \tag{21}$$

其中 A_1 A_2 分别为圆中心和椭圆中心拟合的正弦曲线的振幅。

利用误差传递公式，换算到计算旋转中心时的误差得

$$\frac{\Delta x}{x} = \sqrt{(\frac{\partial \ln x}{\partial d})^2 \times (\frac{\Delta d}{d})^2} = 0.14\% \tag{22}$$

也在可以接受的范围内。

5.4.1.2 稳定性

首先，考虑到有时候角度给的范围会小于 180 度，此时我们的 iRadon 变换算法精度不会受到影响，但是会增加拟合正弦曲线时的误差。并且当所给角度范围很小的时候，由于图像本身的局限性，可能会无法重建几何信息，那就很难对旋转中心的参数进行标定了。但是不管怎么样，对探测器间距以及角度的标定都是很稳定的，并不会随着角度范围大小的改变而改变。

其次是标定模型的摆放对参数标定造成的影响，由于人工摆放的不稳定性，很多给出的参数可能是有较大误差的，比如原图中的椭圆和圆的圆心距，这一距离的误差将直接影响我们对角度的标定。

5.4.2 新模型的建立与求解

我们考虑建立如图 4-1、4-2 的新模型

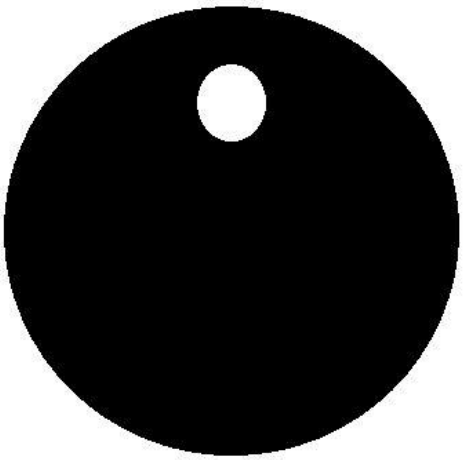

图 4-1 新模型示例图

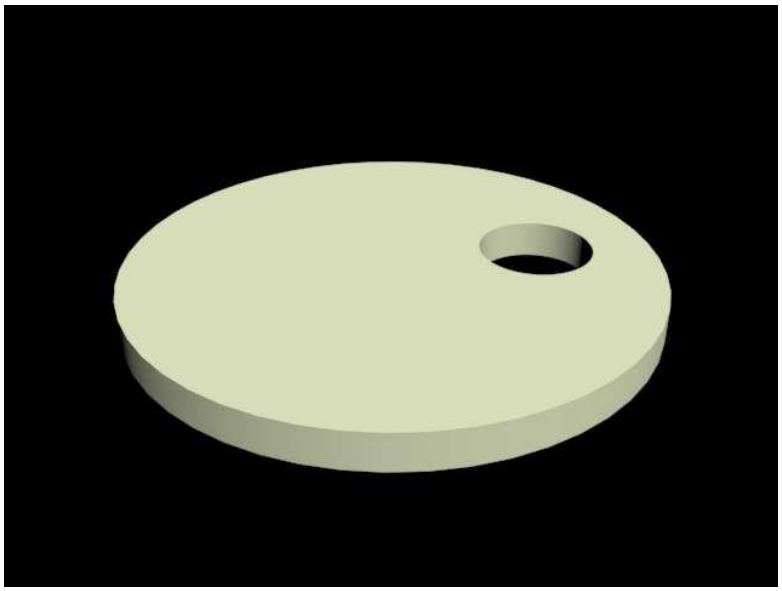

图 4-1 新模型示例立体图

此模型的标定方法与之前类似，也是通过分离大小圆的数据，然后分别拟合各自的中心来得到各自的正弦曲线。

但是不同的是，此时利用 180 个大圆的直径来拟合真正的大圆直径，从而算得探测器单元间距，所用数据会比原方法多很多，所以可以一定程度的提高所得探测器单元间距。

其 radon 变换后的矩阵示意图如图 4-3

图 4-2 新模型的接收信息矩阵

可以看出小圆的数据完全地包含在大圆中，并且完全不需要人工摆放标定，

此时二者圆心距的精度属于工业制造级的精度，完全是可以很精确的。

并且此时数据对角度的均匀性也没有依赖，较少的角度也可以得出精确的标定，仍然拥有更高的稳定性。

6 模型评价

6.1 模型优点

1）问题 1 中的算法利用了附件 2 中几乎所有的数据，有效降低了由某些数据的偶然误差带来的影响。

2）问题 1 中的算法没有默认 X 射线方向角度的变化量，而是在具体求解出每个方向的角度，再进行 180 列间的拟合操作，准确性明显提高。

3）我们的代码很好的消除了因离散化的反 Radon 变换引起的误差，使精度达到非常高的要求。

4）我们自己优化的 FBP 代码，可以自己设置参数，提高了代码的适用性，不单是解决了一个问题，由其兼容性可以延伸到其他问题当中，非常普适。

5）在先仿射变换后反 Radon 变换，与先反 Radon 变换后仿射变换中，我们选择了前者，充分利用了所给的 X 光透射信息，这比后者带来的精度高很多。后者在实际操作中因为仿射变换的旋转放缩等变换对格点的作用不再是格点，会让人对 “反 Radon 变换的重建矩阵中的具体值应该赋予哪个格点”这个问题感到困惑与不安。我们将仿射变换放在反 Radon 变换前可以充分消除这种不安，提高了精度。

6）在第二问的求解中，我们使用了分域平均化的操作使得噪音被消除，即消除了离散化重建所产生的误差。

7 参考文献

【1】 黄力宇, 朱守平, & 匡涛. (2015). *医学断层图像重建仿真实验*. 西安电子科技大学出版社.

8 附录

average.m

```
%=  之  前  必  须  先  运  行  main23.m=====  初  始  化
=======================================%
sum1=0;sum2=0;sum3=0;sum4=0;sum5=0;
count1=0;count2=0;count3=0;count4=0;count5=0;
R=zeros(256,256);
AS3=xlsread('A 题附件.xls','附件 4');
%====================  将  各  个  点  分  类
=======================================%
for i=1:256
    for j=1:256
        z=rec_RL(j,i);
        if(filternumber( i,j,z) ==1)
```

```
            sum1=sum1+z;
            count1=count1+1;

        elseif(filternumber( i,j,z) ==2)
            sum2=sum2+z;
            count2=count2+1;

        elseif(filternumber( i,j,z) ==3)
            sum3=sum3+z;
            count3=count3+1;

        elseif(filternumber( i,j,z) ==4)
            sum4=sum4+z;
            count4=count4+1;

        elseif(filternumber( i,j,z) ==5)
            sum5=sum5+z;
            count5=count5+1;

        end
    end
end
%============ 求 各 类 点 的 均 值 作 为 那 个 面 的 值
==================================%
average1=[sum1/count1 sum2/count2 sum3/count3 sum4/count4 sum5/count5];

for i=1:256
    for j=1:256
        z=rec_RL(j,i);
        if(filternumber( i,j,z) ==1)
            R(j,i)=average1(1);

        elseif(filternumber( i,j,z) ==2)
            R(j,i)=average1(2);
            if(i<120 || i>130 || j<100)
                R(j,i)=average1(1);
            end

        elseif(filternumber( i,j,z) ==3)
            R(j,i)=average1(3);

        elseif(filternumber( i,j,z) ==4)
            R(j,i)=average1(4);
```

```
        elseif(filternumber( i,j,z) ==5)
            R(j,i)=average1(5);

        end
    end
end
%===第二问==分别用不用阈值处理和用阈值处理的方法===找出 10 个点对应的吸收率======%
Points=AS3;
answer=zeros(10,3);
answer1=zeros(10,3);
for k=1:10
    p=Points(k,:);
    temp=2.56*[p(1),100-p(2)];
    answer(k,:)=[p interp2(rec_RL,temp(1),temp(2))];
    answer1(k,:)=[p interp2(R,temp(1),temp(2))];
end

answer
answer1
```

filling.m

```
function [ C ] = filling( A,B )
%%=============结合椭圆得到每列圆的数据函数=============%%

C=A;
for k=1:180
   for n=1:512
       a=B(k,1);
       b=B(k,2);
       c=B(k,3);
       temp=a*n^2+b*n+c;
       if temp>0
           C(n,k)=A(n,k)-sqrt(temp);
           if C(n,k)<0.01
               C(n,k)=0;
           end
       end

   end
end
end
```

filternumber.m

```
function [ z1 ] = filternumber( x,y,z )
%FILTERNUMBER Summary of this function goes here
%   Detailed explanation goes here

if y>120
    if z<14/28
        z1=0;
    else
        z1=1;
    end
else
    if z<14/28
        z1=0;
    elseif z<28.2/28
        z1=1;
    elseif z<32/28
        z1=2;
    elseif z<35/28
        z1=3;
    elseif z<40/28
        z1=4;
    elseif z<1.5
        z1=5;
    end
end
```

main1.m

```
%==============                初                始                化
=================================================%
%AS0=xlsread('A题附件.xls','附件1');
AS1=xlsread('A题附件.xls','附件2');
%AS2=xlsread('A题附件.xls','附件3');
%AS3=xlsread('A题附件.xls','附件4');
%AS4=xlsread('A题附件.xls','附件5');
Ellipse=zeros(180,3);
Circle=zeros(180,3);


%=====先去掉有圆的数据,拟合得到每列椭圆的方程 sqrt(ax^2+bx+c),记录
a,b,c=======%
```

```
for k=1:180
    [A,B]=unzip(k,AS1);
    A=A.^2;
    target=polyfit(B,A,2);
    Ellipse(k,1)=target(1);
    Ellipse(k,2)=target(2);
    Ellipse(k,3)=target(3);
end

%=============== 去除椭圆的数据，留下圆的数据 ==============================%

Circle_sin=filling(AS1,Ellipse);

%=============== 得到每列圆的方程 sqrt(ax^2+bx+c), 记录 a,b,c=================%

for k=1:180
    [A,B]=unzip_circle(k,Circle_sin);
    A=A.^2;
    target=polyfit(B,A,2);
    Circle(k,1)=target(1);
    Circle(k,2)=target(2);
    Circle(k,3)=target(3);
end

%============== 利用圆半径的不变性求探测器间距 ==============================%

diameter=0;
for k=1:180
    a=Circle(k,1);
    b=Circle(k,2);
    c=Circle(k,3);
    diameter=diameter+sqrt(b^2-4*a*c)/abs(a);
end
diameter=diameter/180;
R=8/diameter


%============= 求出椭圆和圆的中心 ======= 顺带求出两中心差 ====================%
```

```
center_Ellipse=zeros(180,1);
center_Circle=zeros(180,1);
center_Distance=zeros(180,1);



for k=1:180
    target=Ellipse(k,:);
    target2=Circle(k,:);
    center_Ellipse(k)=-target(2)/(2*target(1));
    center_Circle(k)=-target2(2)/(2*target2(1));
    center_Distance(k)=-
target(2)/(2*target(1))+target2(2)/(2*target2(1));
end

%==============          求          角          度          值
================================================%
degree=zeros(180,1);
for k=1:180
    degree(k)=180*asin(center_Distance(k)*R/45)/pi;
end
%=========================     修     正     arcsin     函     数
==================================%

temp=1;
min=abs(degree(2)-degree(1));

for k=2:180
    if(abs(degree(k)-degree(k-1))<min)
        min=abs(degree(k)-degree(k-1));
        temp=k;
    end
end
for k=temp:180
    degree(k)=180-degree(k);
end

%================分别拟合出圆和椭圆的中心并拟合对应中心的正弦曲线
==========%
f = fittype('a*sin(b*x+c)+d'); %   拟合函数的形式
fit1 = fit(degree,center_Ellipse,f,'StartPoint',[48,0.01,0.8498,200]);
fit2 = fit(degree,center_Circle,f,'StartPoint',[200,0.01,0.4905,200]);
```

```
%=====利用两正弦曲线振幅求出两中心到旋转中心的距离,并求出旋转中心坐标
=========%

%=================WARNING 坐 标 原 点 为 左 上 角
==================================%

r1=fit2.a*R;%=====R 为 探 测 器 间 距
==========================================%
r2=fit1.a*R;

%================= 旋 转 中 心 坐 标
==========================================%

x=(r2^2-r1^2+95^2-50^2)/90
y=50-sqrt(r1^2-(x-95)^2)

%================== 转 到 iteration.m 开 始 迭 代 操 作
============================%
```

main23.m

```
AS2=xlsread('A 题附件.xls','附件 3');
AS3=xlsread('A 题附件.xls','附件 4');
AS4=xlsread('A 题附件.xls','附件 5');
%================ 仿 真 参 数 设 置
==========================================%
N = 256; % 重建图像大小，探测器采样点数也设为 N
delta = pi/180; % 角度增量(弧度)
theta = degree+90; % 所有投影角度
theta_num = length(theta);
d = 1; % 平移步长
%============ 产 生 滤 波 函 数
==============================================%
fh_RL=Rlfilter(512,d);
%============== 滤 波 反 投 影 重 建 （ 利 用 卷 积 ）
==================================%
rec_RL = RL_FBP(theta_num, N, AS2, delta, fh_RL,theta);
rec_RL2 = RL_FBP(theta_num, N, AS4, delta, fh_RL,theta); % R-L 函数滤
波反投影重建
%=============== 结 果 显 示
==================================================%
imshow(rec_RL,[])
%surf(rec_RL)
```

```
%===== 第 三 问 ======= 找 出  10  个 点 对 应 的 吸 收 率 ====================================%
Points=AS3;
answer=zeros(10,3);
for k=1:10
    p=Points(k,:);
    temp=2.56*[p(1),100-p(2)];
    answer(k,:)=[p,interp2(rec_RL2,temp(1),temp(2))];
end
surf(rec_RL2)
```

RL_FBP.m

```
function  [rec_RL] = RL_FBP(theta_num,N,R1,delta, fh_RL,deg)
% R-L filtered back projection function
% ------------------------------
% 输入参数 ：
% theta_num    ：  投影角度个数
% N        ：  图像大小、探测器通道个数
% R1       :  投影数据矩阵
% delta    :  角度单位
% fh_RL    :  R-L 滤波函数
% 输出参数 ：
% rec_RL  :  反投影重建矩阵
rec_RL = zeros(N);
for m = 1:theta_num
    pm = R1(:,m); %某一角度的投影数据
    pm_RL = conv(fh_RL, pm, 'same');    % 做卷积
    Cm = (N/2)*(1-cos(deg(m)*delta)-sin(deg(m)*delta));
    tx = floor(N/100*(50-40.733651335197870));
    ty = floor(N/100*(50-43.727107987934176));%旋转中心平移像素
        for k1 = 1-ty:N-ty
            for k2 = tx+1:N+tx
                %以下是射束计算，注意射束编号 n 取值范围为 1~N-1
                Xrm        =        (Cm+(k2-1)*cos(deg(m)*delta)+(k1-1)*sin(deg(m)*delta)-N/2)/sqrt(2)+N/2;
                n = floor(Xrm); % 射束编号（整数部分）
                t = Xrm-floor(Xrm); % 小数部分
                n = max(1,n); n = min(n,N-1); % 限定 n 范围为 1~N-1
                p_RL                                =((1-t)*(pm_RL(2*n-1)+pm_RL(2*n))+t*(pm_RL(2*n+1)+pm_RL(2*n+2)))*delta; % 线性内插，由此线性内插限制了此代码只能用于 N/2 的矩阵重建
                rec_RL(N+1-k1-ty,k2-tx)     =     rec_RL(N+1-k1-ty,k2-tx)+p_RL; % 反投影
```

```
            end
        end
end
end
```

Rlfilter.m

```
function  [fh_RL] = Rlfilter(N,d)
% R-L filter function
% ----------------------------
% 输入参数 ：
% N   : 图像大小、探测器通道个数
% d   : 平移步长
% 输出参数 ：
% fh_RL  : R-L 滤波函数
fh_RL = zeros(1,N);
for k1 = 1:N
    fh_RL(k1) = -1/(pi*pi*((k1-N/2-1)*d)^2);
    if mod(k1-N/2-1,2) == 0
        fh_RL(k1) = 0;
    end
end
fh_RL(N/2+1) = 1/(4*d^2);
end
```

unzip.m

```
function [A,B] = unzip( num, M)
%UNTITLED 此处显示有关此函数的摘要
%   此处显示详细说明
C = M(:,num);
A=[];
B=[];
mid = round(-200*sin(2* pi /360*(num-55))+258);
for n=1:512
    if (C(n)~=0 && (n<=mid-25 || n>=mid+25))
        A=[A;C(n)];
        B=[B;n];
    end
end
end
```

unzip_circle.m

```
function [A,B] = unzip_circle( num, M)
%UNTITLED 此处显示有关此函数的摘要
%   此处显示详细说明
C = M(:,num);
A=[];
B=[];

for n=1:512
    if (C(n)~=0)
        A=[A;C(n)];
        B=[B;n];
    end
end
end
```