

CT系统标定与图像重建

摘 要

为解决CT系统参数标定及成像问题，本文由相关理论基础入手，剖析模板形态学特征，以此为切入点建立并优化了标定模型，并且对根据未知介质进行图像重建，根据图形特性实现了噪声去除。

针对问题一，我们根据基础理论知识，结合几何计算，得出标定模型在各个照射角度中的投影强度表达式，尝试通过函数拟合的方法求解。在扫描图中，圆和椭圆的投影函数相互耦合，为了更好的将其区分开，我们从图像处理的角度入手，通过高通滤波和形态学操作大致提取出椭圆曲线边界，利用其宽度信息粗略求出各次投影的角度。再利用粗测的角度，计算出圆的轨迹，挖去受圆形投影影响的数据点，以此来消除圆的投影函数对椭圆投影函数的影响，用剩下的数据对椭圆投影曲线进行拟合，得到了很好的效果，以较高的精度精确计算出X光各次照射角度及其他标定相关的参数。

针对问题二、三，考虑到逆Radon变化的卷积特性，首先使用线性插值得到新的图像作为逆Radon变换的输入，再经由坐标变换，内插缩放、滤波变换等操作得到最终的介质吸收率分布图。

针对问题四，模型检验部分，我们从误差理论角度进行分析，研究标定模型的敏感性，并尝试人为引入噪声，验证模型的稳定性。在研究过程中我们发现了原定标模型在特定角度附近精度大幅下降的问题，针对该问题，尝试设计出“三角形”标定模板等若干种模板，最终兼顾合理性及稳定性，确定“双椭圆”模板作为新标定模板，从理论上证明其稳定性优于原模版。

关键词 CT系统、Radon变换、图像处理、最小二乘拟合、插值、滤波



1 问题重述

1.1 问题背景

X射线是一种能够穿透物体的能量射线，并会产生衰减、折射、散射等物理现象。计算机断层成像(Computed Tomography, CT)技术依据X射线穿过待测物体发生吸收衰减的原理，可以在不破坏物体内部结构的情况下对其进行三维可视化成像。CT技术作为一种重要的无损检测技术，广泛应用于医学成像、工业探伤、货运安检以及文物复原等领域。

CT技术的发展大致经历四个阶段：第一代CT设备通过平行束平移旋转扫描获得投影数据；第二代设备使用小角度扇形射线束代替平行束；第三代设备仅包含扇形束的旋转扫描动作，不再采用平移运动；第四代设备将探测器固定于360°圆周上，仅旋转X射线源以解决环形伪像问题。由于被测物与扫描环境的复杂性，CT扫描方式日趋灵活。

CT技术快速发展，对CT设备的精密度要求随之提升。由于安装过程中存在误差，实际的CT成像系统不满足理想成像关系，会使断层图像的重建质量受到影响。因此，使用CT系统前需要借助已知结构的样品进行标定，修正安装误差。

1.2 相关信息

典型初代二维CT系统如图??所示，探测器平面由512个等距接收点组成，X射线垂直入射。发射器与探测器相对位置固定不变，探测系统绕固定旋转中心逆时针旋转180次。对每一个X射线方向，探测器测量经位置固定不动的二维待检测介质吸收衰减后的射线能量，并经过增益等处理后得到180组接收信息。

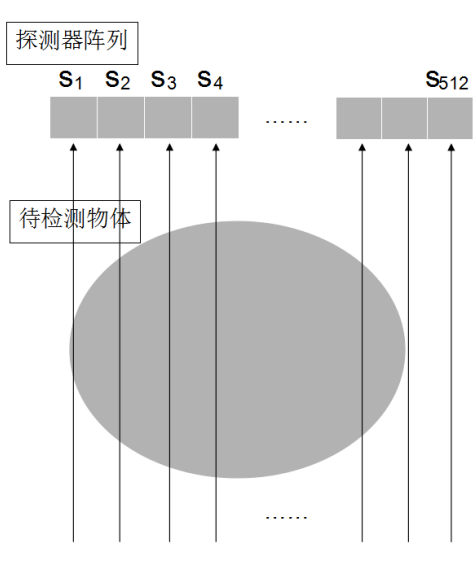


图 1: CT系统示意图



1.3 需解决的问题

本文将题述问题归结为以下三个部分，建立数学模型进行分析研究。

问题一：参数标定

根据标定模板的几何形状与附件1给出的吸收强度，以及附件2给出的探测器接收信息，标定CT系统的相关参数，包括旋转中心的位置、探测器单元间距、X射线的180个投射角度。

问题二、三：CT成像

利用已标定的CT系统参数，结合附件3、附件5给出的未知介质接收信息进行成像，确定未知介质的位置、几何形状与吸收率等信息，并具体给出附件4所给位置的吸收率。

问题四：模型分析与改进

分析问题一中参数标定的精度与稳定性，自行设计新模板、建立对应标定模型以做出改进。

2 问题分析

本问题主要研究CT系统的参数标定与成像。首先通过投影信息确定X射线入射角度及探测器单元间距，通过图像重建确定旋转中心位置，完成系统参数标定；其次由投影信息直接重建图像，获取未知介质信息；最后分析参数标定的精度与稳定性，设计新模板并建立标定模型。

2.1 问题一

分析问题一，建立投影强度曲线的数学模型。首先分析椭圆投影宽度随X射线角度的变化规律，以粗略确定各入射角度及旋转中心。在各入射角度对应的投影强度曲线中，通过图像重建去除圆形模板影响的点位，拟合椭圆投影强度曲线，求解增益系数、探测器单元间距、入射角度的关系。其次在原强度曲线中，扣除椭圆投影的影响，得到180组精确的圆形投影强度曲线。拟合圆形投影强度曲线，通过投影宽度求解探测器单元间距，从而精确求解增益系数与X射线入射角度。最后通过图像重建，确定旋转中心位置。



2.2 问题二、三

CT系统的工作原理是由投影重建图像，针对平行束系统的重建算法包括直接反投影法、滤波反投影法、卷积反投影法等，相关文献中已有详细论述。针对问题二应以投影信息与标定所得系统参数作为输入，通过逆Radon变换重建图像，并对图像进行滤波去噪处理，以得到未知介质的信息。

2.3 问题四

分析问题一中参数标定的精度与稳定性，针对确定 θ 取值时所用的 $\arccos\theta$ 函数在极值点附近对误差的敏感性进行改进。提出以斜 45° 椭圆代替圆形的新模板，建立相应标定模型，对其精度与稳定性进行评估。

3 假设与符号

3.1 模型假设

1. 入射的X射线完全平行，忽略相互干涉。
2. 探测器单元阵列与二维待检测介质处于同一平面中，忽略载物台在任意角度可能发生的倾斜。
3. 附件所给接收信息均为精确值。

3.2 符号说明

4 模型建立与求解

4.1 CT成像的理论基础

X射线与物质的相互作用

当X射线照射在被检测物体上时，一部分射线能量被物体吸收，使得射线强度发生衰减。其衰减遵循Lambert-beer吸收定律[?], 呈指数变化。如图??所示，设X射线初始强度为 I_0 ，穿过厚度为 x ，吸收系数为 μ 的均匀介质后强度变为 I ，则有：

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

$$\mu x = \ln \frac{I_0}{I} \quad (2)$$



符号	符号说明	单位
Δd	探测器单元间距	mm
θ_i	第 <i>i</i> 组X射线的入射方向	rad
a	椭圆半长轴	mm
b	椭圆半短轴	mm
R	圆半径	mm
D	椭圆投影宽度	mm
$f(x, y)$	吸收率函数，表示介质吸收强度的分布	
$g(s, \theta)$	投影函数，表示计算所得的投影强度值， s 为真实投影坐标	
$h(l, \theta)$	采样函数，表示探测器采样数据拟合所得曲线， l 为像素坐标，为真实投影长度的 $1/\Delta l$ 倍	
k	增益系数，探测器接收信息与投影强度的比值	mm^{-1}
k_μ	比例系数，表示由采样信息重建的图像与吸收率数据的比值	

表 1: 符号说明

对于二维平面内的不均匀介质，吸收系数表示为厚度 x 的函数 $\mu(x)$ ，此时Lambert-beer吸收定律表示为：

$$\int \mu(x) dx = \ln \frac{I_0}{I} \quad (3)$$

式??表明，X射线穿过物体后的射线强度与被检测物体的吸收系数相关，CT成像的物理本质就是通过探测器接收的强度信息计算吸收系数 μ 的分布。

数据意义的分析解释

相关文献显示，探测器接收的投影信息应与吸收系数 $\mu(x)$ 沿投影方向的线积分成正比，现有图像重建算法均以此为前提。针对本题所给数据，分析附件1可知标定模板吸收率处处相同；将附件2中同一入射角度对应的全部接收值求和（如图??所示），可见接收值总和基本保持不变。可以认为，本题所给数据存在类似关系，即探测器接收信息与吸收率函数 $f(x, y)$ 沿X射线入射方向的线积分（即投影强度）成正比：

$$\int f(x, y) dl = \frac{1}{k} h(l, \theta) \quad (4)$$

式中 $h(l, \theta)$ 为探测器接收信息， $f(x, y)$ 为介质的吸收率分布， k 为增益系数。



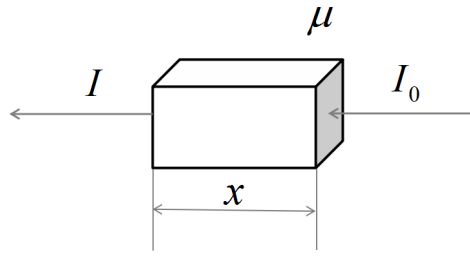


图 2: Lambert-beer吸收定律示意图

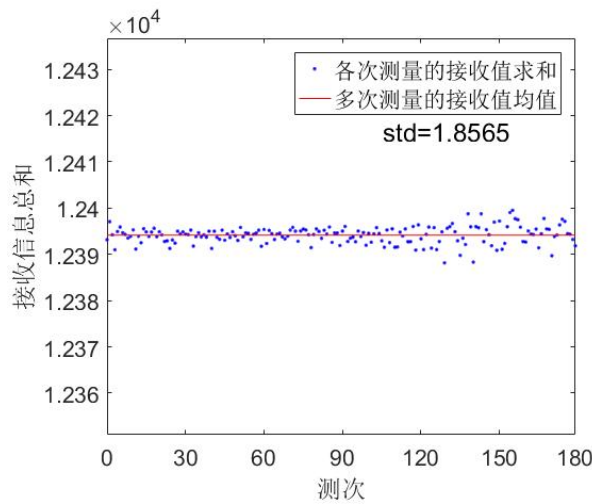


图 3: 各次测量的接收值总和

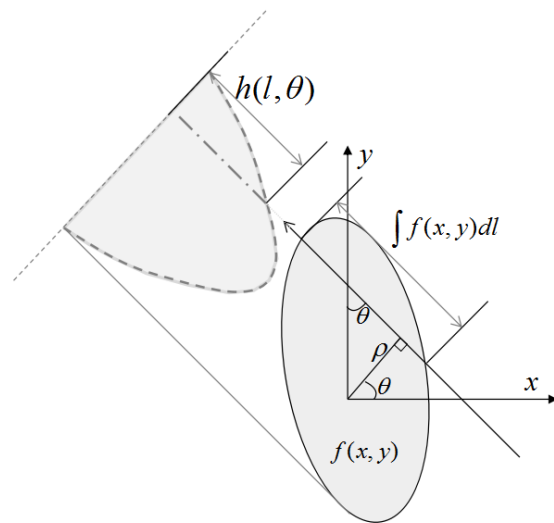


图 4: 接收信息分析示意图

Radon变换

吸收率函数 $f(x, y)$ 沿 θ 方向的投影函数为 $h(\theta, l)$ (如图??所示), 由 θ, l 组成的极坐标系统张成Radon空间, 空间中任意点 (l, θ) 的值实际上代表吸收率函数的一个线积分值。

物体空间的吸收率函数 $f(x, y)$ 与Radon空间的投影函数 $g(l, \theta)$ 有明确的映射关系, 通常将物体空间函数变换至Radon空间函数的过程称为Radon变换, 反之称为逆Radon变换。实际上, Radon空间的函数值是CT系统中探测器采集的一个数据点, 只要有充分的扫描数据, 就可以通过插值方式重建物体空间的图像。

4.2 问题一

首先建立投影强度模型, 分析强度曲线线型的影响因素, 可知标定参数的核心问题在于入射角度的确定。模型求解部分从数据图像的形态学角度出发, 粗略确定X射线的入射角度, 从而分离椭圆与圆形模板的投影强度, 通过拟合投影函数标定CT系统各参数值。



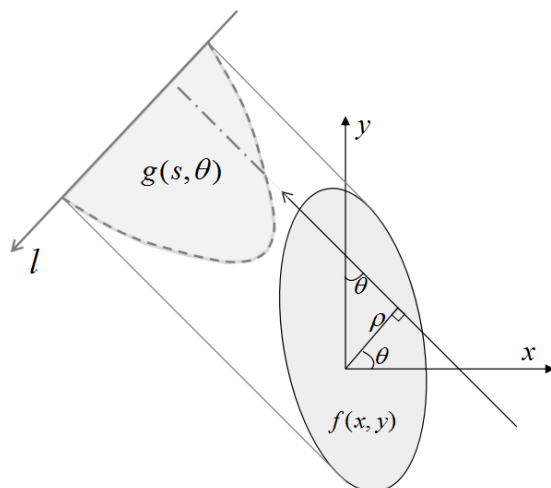


图 5: 吸收率函数及其投影函数

4.2.1 投影强度模型的建立

记第 i 组X射线入射方向与正方形托盘竖直轴的夹角为 θ_i ，探测器位于托盘正下方时 θ 为0，以逆时针方向为正。记旋转中心为 O 点，在探测器平面的投影点为 O_θ ，以此为原点建立投影强度坐标轴（如图??所示），表示物体空间的真实投影坐标，单位为 mm 。记椭圆中心为 O_1 ，圆形中心为 O_2 ， $\angle O_1 O O_\theta = \alpha_1$ ， $|O O_1| = A_1$ ， $\angle O_2 O O_\theta = \alpha_2$ ， $|O O_2| = A_2$ ，则 O_1 ， O_2 在 S_θ 轴上的投影为：

$$s_{o_1}(\theta) = A_1 \sin(\alpha_1 - \theta) \quad (5)$$

$$s_{o_2}(\theta) = A_2 \sin(\alpha_2 - \theta) \quad (6)$$

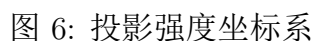
首先建立椭圆的投影强度模型。以椭圆中心为基准，计算吸收率的线积分值可得：

$$g'_1(s', \theta) = \sqrt{-\frac{4a^2b^2}{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)^2} s'^2 + \frac{4a^2b^2}{a^2 \sin^2 \theta + b^2 \cos^2 \theta}} \quad (7)$$

于是在 s_θ 轴中，有

$$\begin{aligned} g_1(s, \theta) &= g'_1(s - s_{o_1}, \theta) \\ &= \sqrt{-\frac{4a^2b^2}{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)^2} (s - s_{o_1})^2 + \frac{4a^2b^2}{a^2 \sin^2 \theta + b^2 \cos^2 \theta}} \end{aligned} \quad (8)$$




$$\begin{aligned}
h_1(l, \theta) &= k g_1(l \Delta d, \theta) \\
&= \sqrt{-\frac{4a^2 b^2 k^2 \Delta d^2}{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)^2} \left(s - \frac{s_{o1}}{\Delta d}\right)^2 + \frac{4a^2 b^2 k^2}{a^2 \sin^2 \theta + b^2 \cos^2 \theta}}
\end{aligned} \tag{9}$$
$$K_1 = -\frac{4a^2b^2k^2\Delta d^2}{(a^2\sin^2\theta + b^2\cos^2\theta)^2}$$

$$C_1 = \frac{4a^2b^2k^2}{a^2\sin^2\theta + b^2\cos^2\theta}$$
$$h_1^2(l) = K_1(l - \frac{s_{o1}}{\Delta d})^2 + C_1 \quad (10)$$
$$\begin{aligned} K_2 &= -4k^2 \Delta d^2 \\ C_2 &= 4R^2 k^2 \end{aligned}$$

$$h_2^2(l) = K_2(l - \frac{s_{o2}}{\Delta d})^2 + C_2 \quad (11)$$

4.2.2 投影强度模型的求解

形态学粗测入射角度、旋转中心以及探测器单元间距

椭圆模板的投影宽度 D 可由X射线入射角度唯一确定（如图??所示），其计算公式为[?]:

$$D = \frac{2\sqrt{m^2a^2 + b^2}}{\sqrt{m^2 + 1}} \quad (12)$$

其中 m 为切线斜率，由此得到入射角度的计算公式为:

$$\theta = \operatorname{arccot} \sqrt{\frac{4a^2 - d^2}{d^2 - 4b^2}} \quad (13)$$

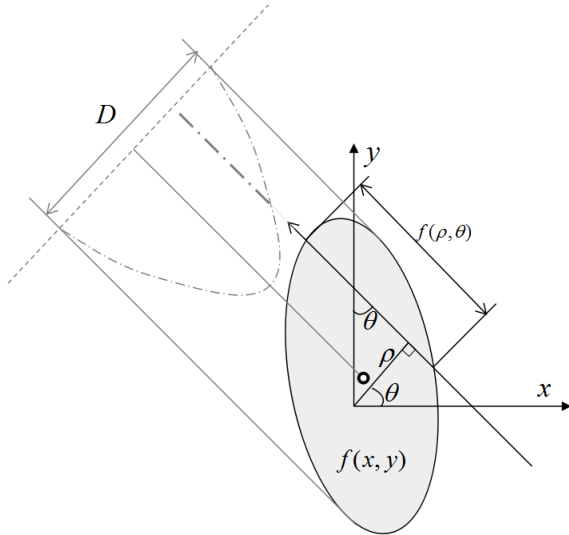


图 7: 椭圆投影宽度示意图

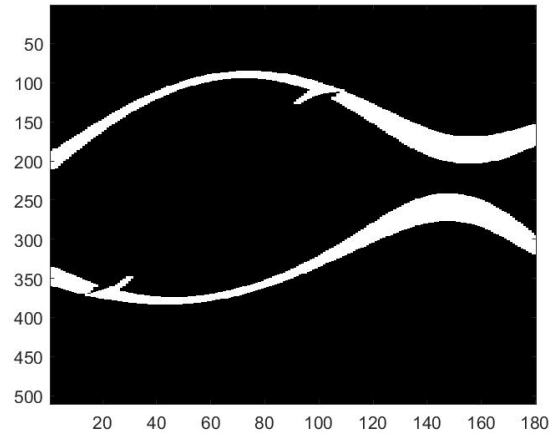


图 8: 椭圆投影宽度粗测用图

通过分析探测器接收信息，将椭圆投影像素宽度的极大值作为长轴的估计值，估算探测器单元间距 Δd 以及每一投影角度对应的真实投影宽度，从而粗略测算180个入射角度的取值。

求解结果如图??所示，可近似认为入射角度线性增加。以等距分布的入射角对探测器接收信息作逆Radon变换，所得图像的中心位置即为旋转中心。以椭圆中心、圆形中心与旋转中心为顶点构造三角形，利用其在两幅图像中的相似性即可粗略确定旋转中心在正方形托盘中的位置。

分离提取投影强度曲线

圆形投影中心位置的确定 粗略测算入射角度及旋转中心后，可估算每一强度曲线中



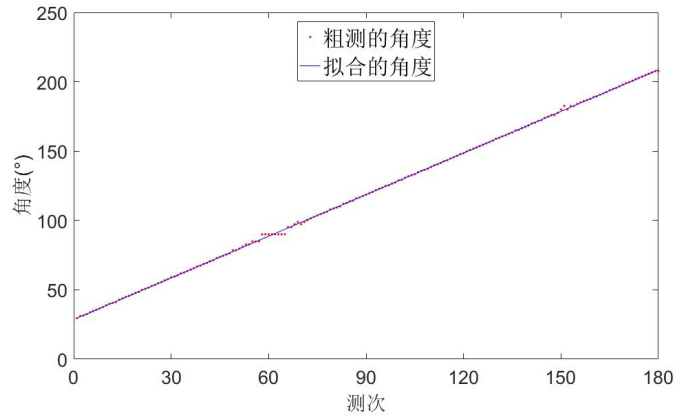


图 9: 入射角度粗测数据图

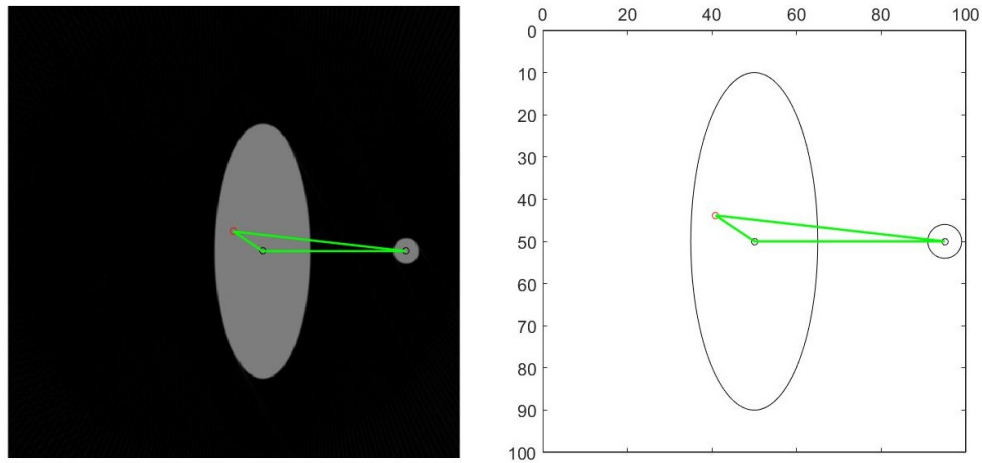


图 10: 重建图像（左）与原始图像（右）

圆形投影的中心位置。

如图??所示，圆心投影位置应满足关系式：

$$l = l_0 \cos(\theta + \alpha) + L_0 \quad (14)$$

式中 l_0 为旋转中心至圆心的像素长度， α 为图??中三角形的一个内角，均可由重建图像估算。 L_0 为旋转中心对应的投影像素坐标，可将椭圆投影宽度的极小点近似作为 $\theta = 0$ 的情形进行估算。

模板投影强度曲线的分离拟合 借助 Δd 估计值可计算圆形投影的像素宽度。在各投影强度曲线中，以圆心投影坐标为中心，在左右两侧去除两倍圆形投影宽度内的数据点，从而完全消除圆形投影的影响，得到椭圆投影强度的部分数据。



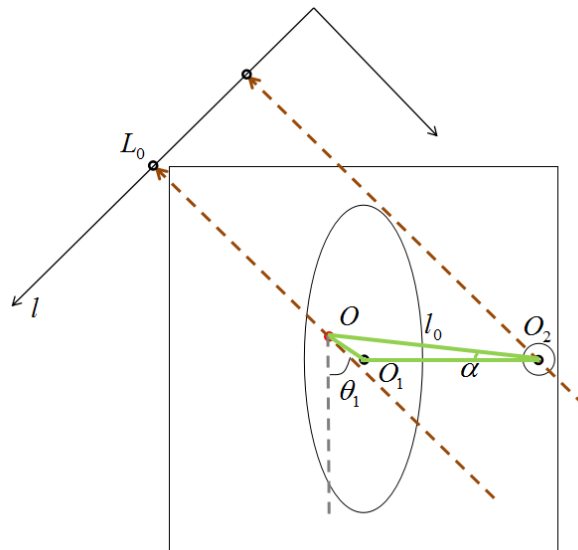


图 11: 圆心投影位置

对各投影角度，利用上述数据拟合二次多项式得到完整的椭圆投影强度曲线。从全部接收信息中扣除椭圆投影强度，即可得到圆形投影强度数据，进而拟合得圆形投影强度曲线。图??显示分离出的投影强度数据，图??显示拟合所得投影强度曲线。

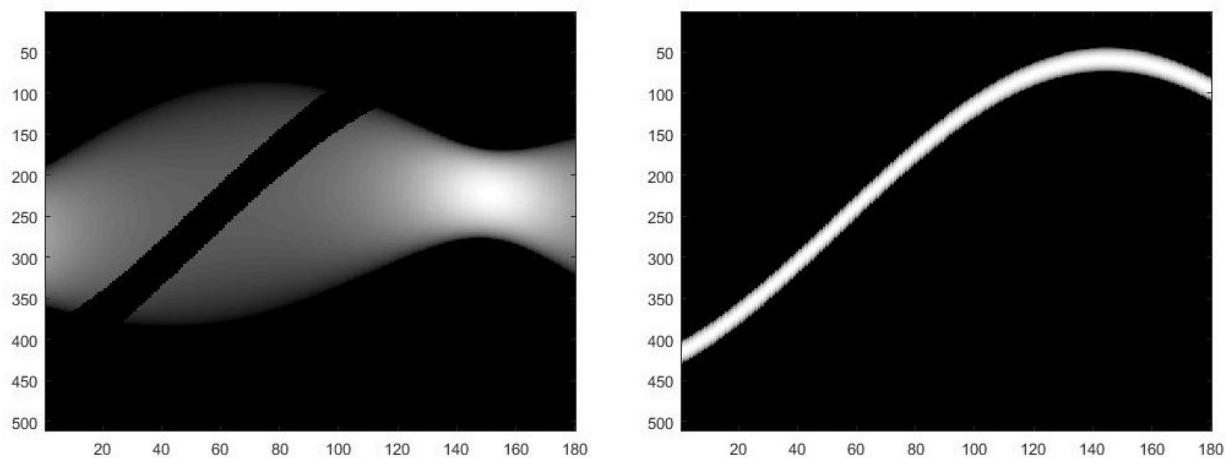


图 12: 椭圆投影强度数据（左）与圆形投影强度数据（右）

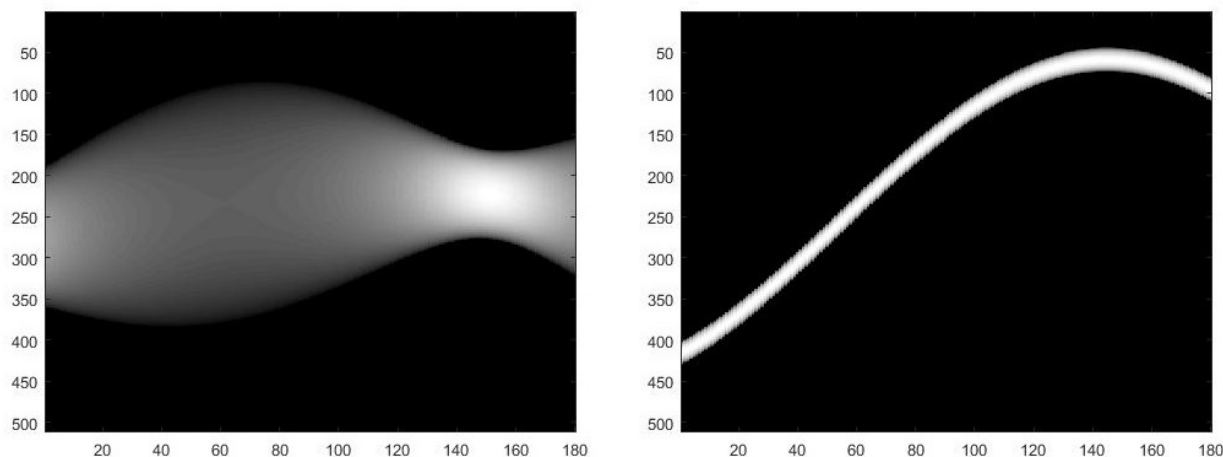


图 13: 椭圆投影强度拟合结果（左）与圆形投影强度拟合结果（右）

标定系统参数

探测器单元间距 Δl 设拟合所得圆形投影强度曲线为:

$$h_2^2(l, \theta) = q_1 l^2 + q_2 l + q_3 \quad (15)$$

与投影模型比较系数得:

$$K_2 = q_1 \quad (16)$$

$$\frac{s_{o2}}{\Delta d} = -\frac{q_2}{2q_1} \quad (17)$$

$$C_2 = q_3 - \frac{q_2^2}{4q_1} \quad (18)$$

则

$$\begin{aligned} -\frac{K_2}{C_2} &= -\frac{q_1}{q_3 - \frac{q_2^2}{4q_1}} = \frac{\Delta d^2}{R^2} \\ \Rightarrow \Delta d &= \sqrt{R^2 \frac{q_1}{\frac{q_2^2}{4q_1} - q_3}} \\ &= 2R \sqrt{\frac{q_1^2}{q_2^2 - 4q_1 q_3}} \end{aligned} \quad (19)$$

拟合180组圆形投影强度曲线，将曲线系数代入式??解出 Δd ，对计算结果取均值得:

$$\Delta d = 0.2768 \text{ mm} \quad (20)$$



关注数学模型
获取更多资讯

入射角度 θ 设拟合所得椭圆投影强度曲线为：

$$h_1^2(l, \theta) = p_1 l^2 + p_2 l + p_3 \quad (21)$$

相应地有：

$$K_1 = p_1 \quad (22)$$

$$\frac{s_{o1}}{\Delta d} = -\frac{p_2}{2p_1} \quad (23)$$

$$C_1 = p_3^2 - \frac{p_2^2}{4p_1} \quad (24)$$

为求出 θ ，对 K_1 ， C_1 表达式进行变换可得：

$$\begin{aligned} -\frac{C_1}{K_1} &= \frac{a^2 \sin^2 \theta + b^2 \cos^2 \theta}{\Delta d^2} = -\frac{p_3^2 - \frac{p_2^2}{4p_1}}{p_1} \\ &= -\frac{p_3^2 - \frac{p_2^2}{4p_1}}{p_1} \Delta d^2 + 2a^2 - b^2 \\ \Rightarrow \cos 2\theta &= -\frac{p_1}{2(b^2 - a^2)} \\ &= -\frac{4(2a^2 - b^2)p_1^2 + (4p_3^2 p_1 - p_2^2) \Delta d^2}{8p_1^2(b^2 - a^2)} \end{aligned} \quad (25)$$

由于探测系统逆时针旋转， θ 值单调增加，由此可确定 θ 的180组取值。

由图??可见，经计算修正后 θ_i 序列更接近其线性拟合值，相邻两点间距 1° 左右，仅在个别点出现偏差。计算结果见表????。

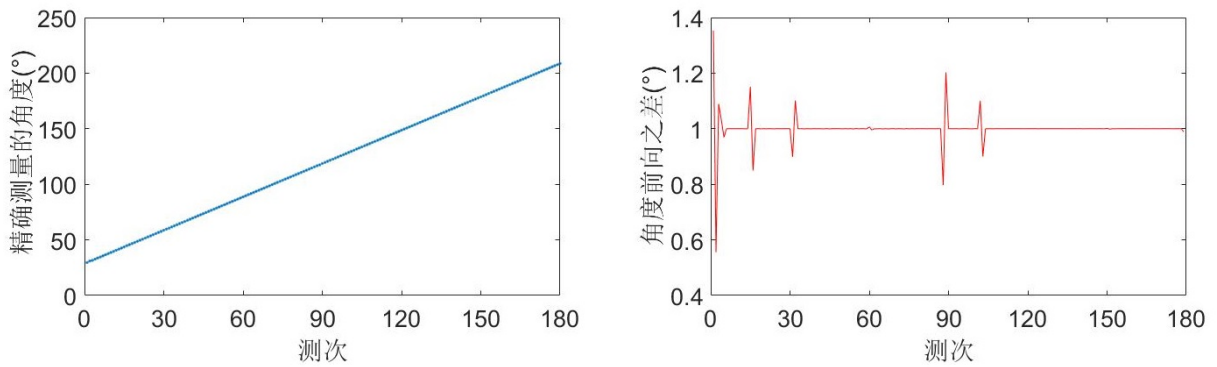


图 14: 入射角度序列 θ （左）与前向对比图（右）



i	$\theta_i/^\circ$	i	$\theta_i/^\circ$	i	$\theta_i/^\circ$
1	29.6463	2	30.9999	3	31.5553
4	32.6447	5	33.6771	6	34.6463
7	35.6463	8	36.6463	9	37.6463
10	38.6463	11	39.6462	12	40.6462
13	41.6463	14	42.6462	15	43.6462
16	44.7967	17	45.6463	18	46.6463
19	47.6464	20	48.6462	21	49.6463
22	50.6460	23	51.6463	24	52.6463
25	53.6461	26	54.6463	27	55.6463
28	56.6462	29	57.6463	30	58.6463
31	59.6463	32	60.5453	33	61.6462
34	62.6462	35	63.6463	36	64.6463
37	65.6462	38	66.6463	39	67.6460
40	68.6462	41	69.6461	42	70.6463
43	71.6462	44	72.6461	45	73.6464
46	74.6458	47	75.6462	48	76.6462
49	77.6462	50	78.6463	51	79.6461
52	80.6463	53	81.6461	54	82.6463
55	83.6456	56	84.6463	57	85.6460
58	86.6463	59	87.6466	60	88.6458
61	89.6520	62	90.6474	63	91.6474
64	92.6470	65	93.6474	66	94.6467
67	95.6467	68	96.6463	69	97.6462
70	98.6467	71	99.6461	72	100.6465
73	101.6464	74	102.646	75	103.6464
76	104.6460	77	105.6463	78	106.6459
79	107.6462	80	108.6462	81	109.6462
82	110.6463	83	111.6462	84	112.6463
85	113.6463	86	114.6463	87	115.6463
88	116.6462	89	117.4436	90	118.6462



i	$\theta_i/^\circ$	i	$\theta_i/^\circ$	i	$\theta_i/^\circ$
91	119.6463	92	120.6462	93	121.6463
94	122.6463	95	123.6463	96	124.6463
97	125.6463	98	126.6464	99	127.6463
100	128.6461	101	129.6462	102	130.6462
103	131.7462	104	132.6463	105	133.6463
106	134.6462	107	135.6463	108	136.6461
109	137.6462	110	138.6463	111	139.6463
112	140.6462	113	141.6463	114	142.6462
115	143.6463	116	144.6462	117	145.6463
118	146.6462	119	147.6463	120	148.6462
121	149.6462	122	150.6463	123	151.6462
124	152.6463	125	153.6462	126	154.6462
127	155.6462	128	156.6463	129	157.6463
130	158.6463	131	159.6463	132	160.6463
133	161.6463	134	162.6463	135	163.6463
136	164.6462	137	165.6463	138	166.6463
139	167.6463	140	168.6463	141	169.6463
142	170.6462	143	171.6462	144	172.6462
145	173.6462	146	174.6463	147	175.6462
148	176.6462	149	177.6465	150	178.6463
151	179.6478	152	180.6463	153	181.6463
154	182.6461	155	183.6462	156	184.6463
157	185.6462	158	186.6463	159	187.6462
160	188.6463	161	189.6462	162	190.6463
163	191.6462	164	192.6463	165	193.6463
166	194.6462	167	195.6463	168	196.6463
169	197.6463	170	198.6463	171	199.6462
172	200.6462	173	201.6463	174	202.6462
175	203.6463	176	204.6462	177	205.6462
178	206.6462	179	207.6463	180	208.6358

旋转中心位置 (x, y) 解得入射角度序列 θ_i 后, 通过线性插值将探测器接收信息处理为入射角度间距相等的情况, 以满足逆Radon算法的输入要求, 获取更高质量的重建图像。



再次运用图??所示的方法，利用三角形相似性确定更精确的旋转中心位置，计算结果如图??所示：

$$x = 40.8436 \text{ mm}$$

$$y = 56.1495 \text{ mm}$$

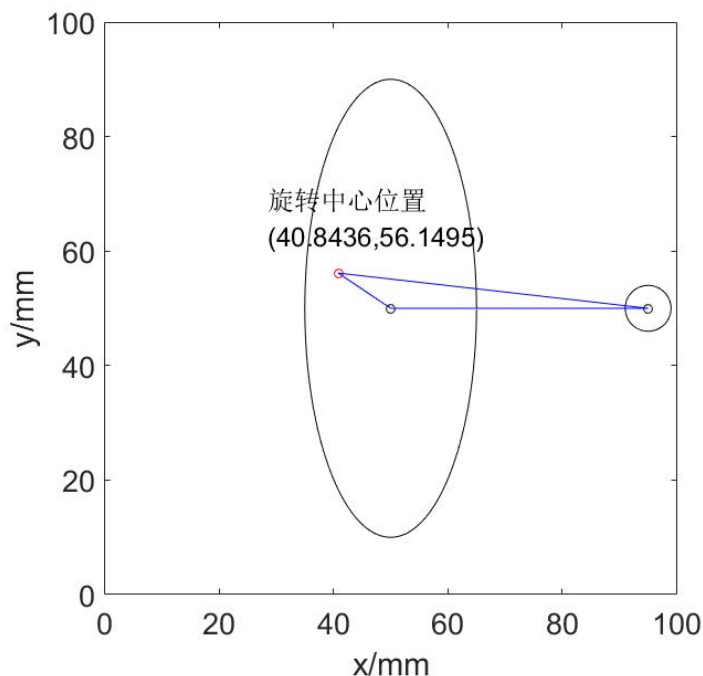


图 15: 旋转中心位置

4.2.3 模型分析与评价

本模型根据拟合结果求解 θ ，由于拟合过程中可能存在误差，故求得的结果也存在一定的误差。

依照前面的推导，有

$$\cos^2 \theta = \frac{1}{b^2 - a^2} \left(-\frac{C}{K} \Delta l^2 - a^2 \right) \quad (26)$$

对于标定用的模型，我们认为所提供的参数值是足够精确的，可以忽略其误差。于是误差的主要来源为 C 、 K 和 Δl 。记参数 P 的误差为 $\delta(P)$ ，则有

$$\delta(\cos^2 \theta) = -\frac{1}{b^2 - a^2} \delta\left(\frac{C}{K} \Delta l^2\right) \quad (27)$$



于是，只需求出 $\delta(\frac{C}{K}\Delta l^2)$ ，根据误差传递公式，有

$$\delta(\frac{C}{K}\Delta l^2) = \delta(\frac{C}{K})\Delta l^2 + \frac{C}{K}\delta(\Delta l^2) \quad (28)$$

其中

$$\delta(\frac{C}{K})/\frac{C}{K} = \sqrt{(\frac{\delta(C)}{C})^2 + (\frac{\delta(K)}{K})^2} \quad (29)$$

因为，K与C都是由二次曲线拟合结果计算得出的，于是分析二次函数拟合参数的置信区间。最小二乘法做二次拟合时，记预测值为 $P(x)$ ，预测残差平方和为 $Q(a_0, a_1, a_2)$ ，则有

$$P(x) = a_2x^2 + a_1x + a_0 \quad (30)$$

$$Q(a_0, a_1, a_2) = \sum_{x=1}^N (P(x_i) - y_i)^2 \quad (31)$$

利用最小二乘的思想进行拟合，则要求 $Q(a_0, a_1, a_2)$ 取最小值，即

$$\begin{cases} \frac{dQ}{da_0} = 0 \\ \frac{dQ}{da_1} = 0 \\ \frac{dQ}{da_2} = 0 \end{cases} \quad (32)$$

计算后求得最优预测为：

$$\begin{bmatrix} m & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix} \quad (33)$$

记

$$M = \begin{bmatrix} m & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \quad (34)$$

则，考虑误差后，有

$$M \begin{bmatrix} a_0 + \Delta a_0 \\ a_1 + \Delta a_1 \\ a_2 + \Delta a_2 \end{bmatrix} = \begin{bmatrix} \sum (y_i + \Delta y_i) \\ \sum x_i (y_i + \Delta y_i) \\ \sum x_i^2 (y_i + \Delta y_i) \end{bmatrix} \quad (35)$$

与原式相减可得：

$$M \begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \Delta a_2 \end{bmatrix} = \begin{bmatrix} \sum \Delta y_i \\ \sum x_i \Delta y_i \\ \sum x_i^2 \Delta y_i \end{bmatrix} \quad (36)$$



即

$$\begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \Delta a_2 \end{bmatrix} = M^{-1} \begin{bmatrix} \sum \Delta y_i \\ \sum x_i \Delta y_i \\ \sum x_i^2 \Delta y_i \end{bmatrix} \quad (37)$$

因为 y_i 为测量误差，可以视为均值为0，标准差为 δ 的独立高斯分布，则有

$$E(y_i) = 0 \quad (38)$$

$$E(y_i y_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (39)$$

至此，可从理论上求得 a_0, a_1, a_2 的表达式，且式子均由 y_i 的线性表达式构成，利用期望的线性性，可以求得 $E(a_0^2), E(a_1^2), E(a_2^2)$ 若对于每一次拟合，都进行如此计算，需要极大的计算量，且不利于直观理解，在此给出一种十分粗糙的方法计算其误差，可以较直观的看到误差的大小：

对于 $y = a_2 x^2 + a_1 x + a_0$ ， y 的相对误差是由 a_2, a_1, a_0 的相对误差共同影响的，由于拟合出来的曲线并非奇异的二次曲线，可以认为 $a_2 x^2, a_1 x, a_0$ 三项的值大小相当，于是 y, a_2, a_1, a_0 的相对误差大致相等，对于 y 的误差，可用其方差的无偏估计来表示，即：

$$\delta(y) = \sqrt{\frac{\sum_{k=1}^N (y_i - \hat{y}_i)^2}{N-1}} \quad (40)$$

y 的值用其均值 \bar{y} 表示，即 y 的相对误差为 $\frac{\delta(y)}{\bar{y}}$ 。如图??所示，用Matlab取出其中一条进行计算，可知该值大致为 $10^{-7} - 10^{-6}$ 。

对于 $\delta(\Delta l)$ ，其值的是利用圆的拟合曲线求解的，单次误差求法与椭圆类似。由于在180行数据中都求得了该值，并取了平均，其误差较之原来减小了 $\sqrt{180} = 13.41$ 倍，与椭圆的误差相比可以忽略。

综上， $\cos^2 \theta$ 的相对误差大概在 10^{-7} 至 10^{-6} 次方量级。因为

$$\frac{d(\cos^2 \theta)}{d\theta} = \sin 2\theta \quad (41)$$

所以

$$\delta(\theta) = \frac{\delta(\cos^2 \theta)}{\sin 2\theta} \quad (42)$$

当 $\sin 2\theta > 10^{-2}$ ，即 θ 与 $\sin 2\theta$ 的零点距离大于0.02时即可以认为，算得的 θ 相对误差小于 10^{-4} 。计算可知，除了极少数几个点之外，其余角度值都可以认为足够精确，相对误差小于 10^{-4} 。



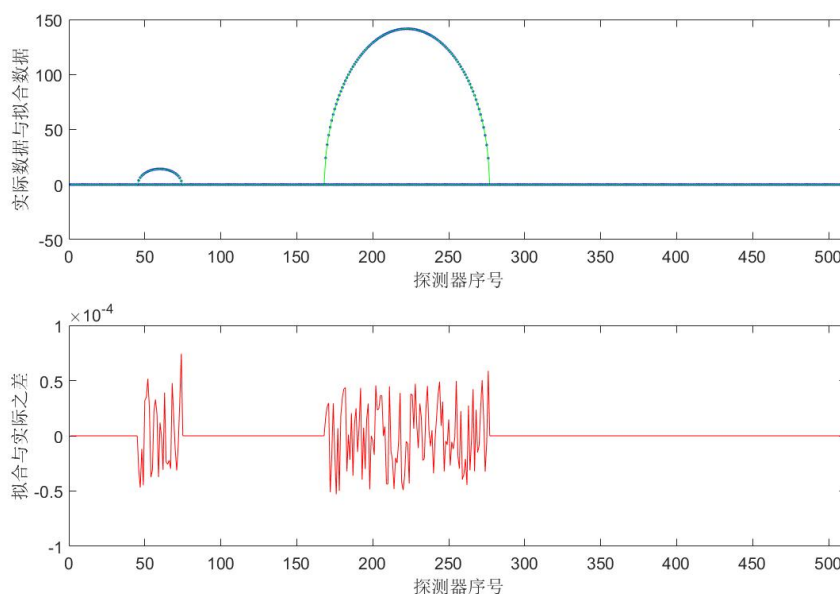


图 16: 曲线拟合及误差计算

4.3 问题二、三

4.3.1 逆Radon变换与图像重建模型

参数修正 本部分将逆Radon变换作为重建CT图像的基础手段。为满足该函数对输入参数的要求，首先将入射角度修正为线性拟合值，通过插值方法对接收信息进行相应修正：

记 $H(k, \theta_i)$ 为入射角度 θ_i 对应的第 k 个采样值，通过线性插值的办法得到角度 $\phi_n = n\alpha$ 处的修正采样函数 $\hat{H}(k, \phi_n)$ 。

设 $\phi_n \in [\theta_i, \theta_{i+1}]$ ，插值方法为：

$$\hat{H}(k, \phi_n) = \frac{1}{\theta_{i+1} - \theta_i} [(\phi_n - \theta_i)H(k, \theta_{i+1}) + (\theta_{i+1} - \phi_n)H(k, \theta_i)] \quad (43)$$

图像重建 以修正后的接收信息与入射角度序列作为输入，通过逆Radon变换重建图像，计算结果如图??所示。

系数修正 由于题述接收信息经由逆Radon变换所得图像数据与吸收率数据不符，两者大致存在线性关系，在此需引入比例系数 k_μ 进行修正。

k_μ 的取值由问题一所给数据计算。观察重建图像断面数据（图??）与全部图像数据的频数直方图（图??），由于标定模板为单一均匀介质，可利用聚类分析算法将图像数据分为两类，对其中高值类取均值即可得到 k_μ 的取值，将重建图像数据除以 k_μ 作为修正。



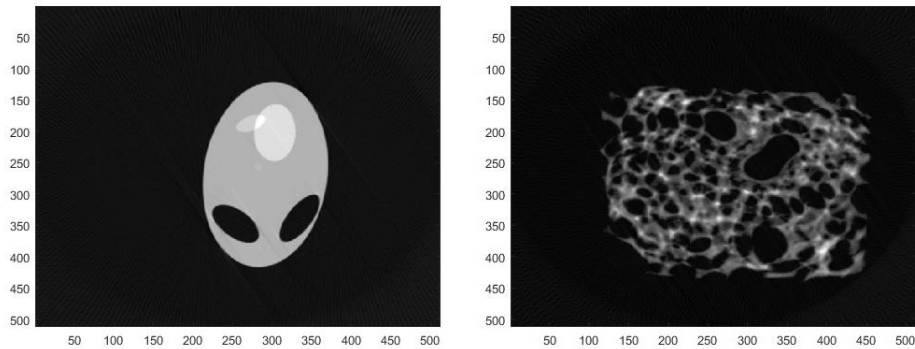


图 17: 未知介质重建图像

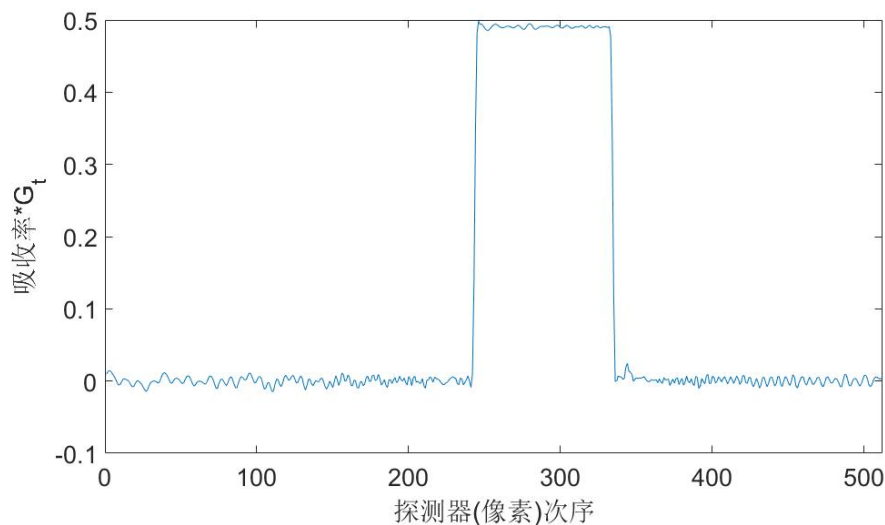


图 18: 重建图像断面数据

滤波去噪 针对图像噪声，利用Wiener滤波进行去噪处理，计算结果如图??-??所示。

对于问题二（附件3），Wiener滤波对几何伪影的消除效果显著。由信号强度频数直方图可见，原始图像的信号强度有若干明显峰值，且峰值附近的分布情况与问题一中的噪声分布类似，滤波去噪使峰型更为尖锐，对噪声有较好抑制效果。因此对于问题二，采用Wiener滤波后的图像数据求解吸收率分布。

对于问题三（附件5），由信号强度频数直方图与滤波前后对比图可见，介质本身吸收率分布广、高频分量大，滤波会抹除较多的边缘信息。因此对于问题三，舍弃Wiener滤波所得结果，利用原始图像求解吸收率分布。

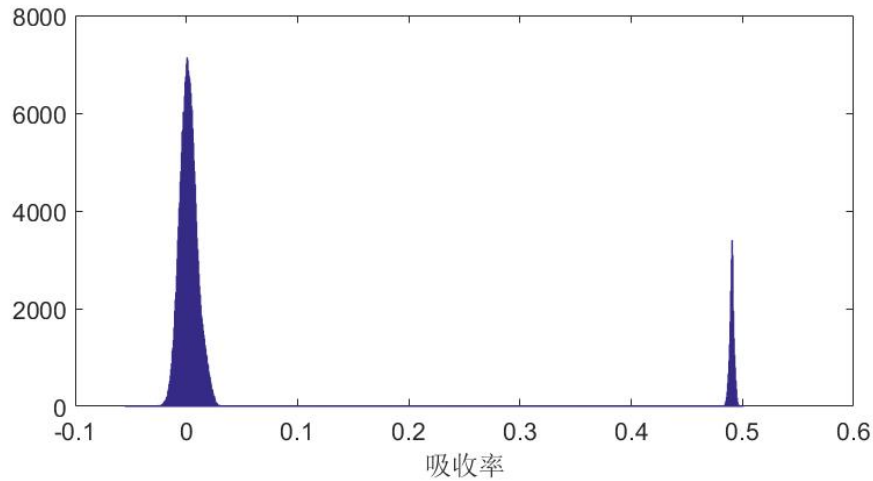


图 19: 图像数据频数直方图

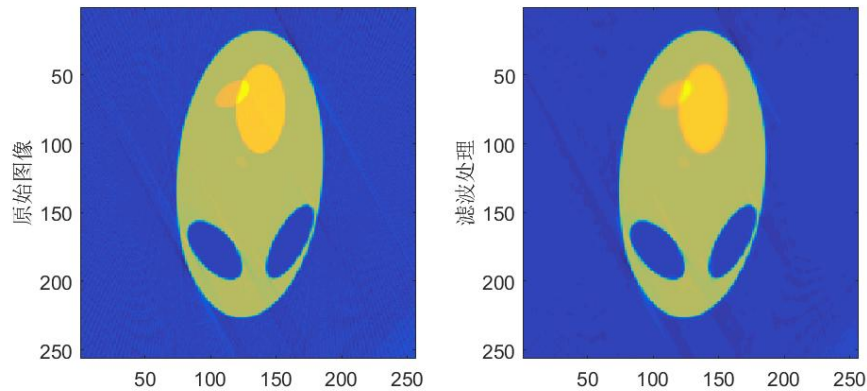


图 20: 问题二重建图（左）与滤波所得结果（右）

4.3.2 未知介质信息的求解

吸收率分布 由于上述重建图像的单位坐标长度对应于探测器阵列的间距，图像数据点位与需求的吸收率分布点位(256×256)并不相符。为提高吸收率求解的精度，应将所求点位变换为重建图像中的坐标，变换方法如下：

$$aa \tag{44}$$

由变换所得坐标利用周围四点数据进行插值运算，即可得到吸收率的分布（如图????所示），计算结果录于附件problem2.xls。对于附件4要求的10个位置，同样运用上述方法计算吸收率，所得结果如下：

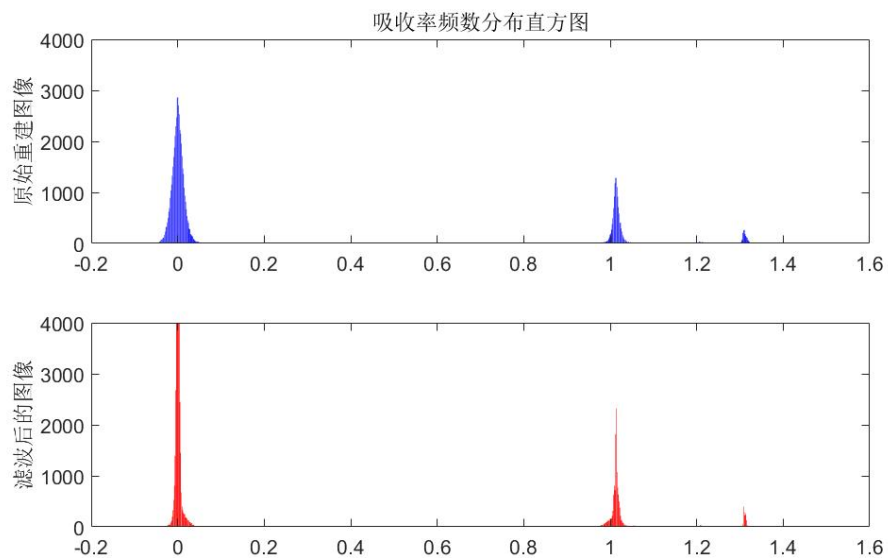


图 21: 问题二投影强度频数直方图（上）与滤波所得结果（下）

位置坐标/ mm	吸收率	位置坐标/ mm	吸收率
(10.0000, 18.0000)	0.0003	(34.5000, 25.0000)	1.0033
(43.5000, 33.0000)	0.0002	(45.0000, 75.5000)	1.2096
(48.5000, 55.5000)	1.0632	(50.0000, 75.5000)	1.4210
(56.0000, 76.5000)	1.3133	(65.5000, 37.0000)	-0.0012
(79.5000, 18.0000)	-0.0043	(98.5000, 43.5000)	0.0012

表 2: 问题二吸收率数据

位置坐标/ mm	吸收率	位置坐标/ mm	吸收率
(10.0000, 18.0000)	0.0657	(34.5000, 25.0000)	2.8779
(43.5000, 33.0000)	6.9510	(45.0000, 75.5000)	-0.0351
(48.5000, 55.5000)	0.2859	(50.0000, 75.5000)	3.2544
(56.0000, 76.5000)	6.3600	(65.5000, 37.0000)	0.0234
(79.5000, 18.0000)	7.2257	(98.5000, 43.5000)	0.0436

表 3: 问题三吸收率数据

介质几何信息 将重建图像坐标变换为正方形托盘坐标，即可得到未知介质的位置与几何形状，如图??所示。



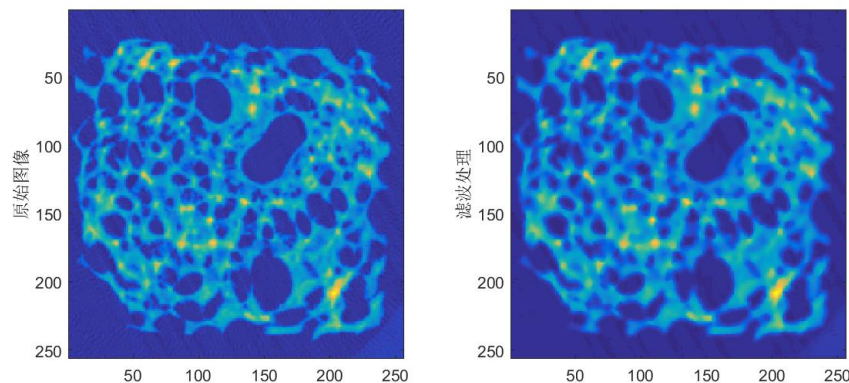


图 22: 问题三重建图（左）与滤波所得结果（右）

4.3.3 模型分析与评价

对于Radon逆变换，依据卷积反投影法，可以先将探测曲线与响应函数进行卷积，之后再行反投影操作，即可得到吸收率的空间分布情况，在连续且理想的情况下，此种方法可以精确的还原出吸收率空间分布图像。但是在实际操作中，由于测量误差及离散化操作，难免会造成一系列误差。主要来源为：

1、探测器采样的误差，可以认为是高斯白噪声，几乎无法消除；

2、投影函数 $f(t)$ 经过采样得到采样函数后造成的信息丢失，利用抽样值复原 $f(t)$ 曲线时，根据采样定理，若 $f(t)$ 的频域表达式 $F(\omega)$ 满足 $F(\omega) = 0 \quad \forall |\omega| > \pi$ ，则可以通过采样函数与 $\frac{\sin(2\pi l)}{l}$ 的卷积来精确还原。但是由于 $f(t)$ 在空间域中的宽度是有限的，由傅立叶变换的性质可知其在频域是无穷宽的，故无法精确复原。其次，通过与 $\frac{\sin(2\pi l)}{l}$ 来复原函数，需要计算到无穷远，不现实，一般采用矩形窗、汉明窗等宽度有限的函数作为替代。于是，量化步骤会丢失一定量的信息。

3、系统函数的傅立叶变换 $C(R)$ 实际上是不存在的，只能用趋近来表示：

$$C(R) = \lim_{\epsilon \rightarrow 0} \frac{2(\epsilon^2 - 4\pi^2 R^2)}{(\epsilon^2 + 4\pi^2 R^2)^2} \quad (45)$$

可以看到 $C(0) = \lim_{\epsilon \rightarrow 0} \epsilon^2 \rightarrow 0$ ，趋于无穷大，实际计算中不可实现，故 ϵ 只能取有限值。为了拟合 ϵ 有限的情况，常常采用“R-L”滤波函数或者“S-L”滤波函数来实现。在这个过程中，也会造成误差。

综上，在进行Radon逆变换的过程中，无可避免会引入误差，需通过平均等方法才能减小误差。



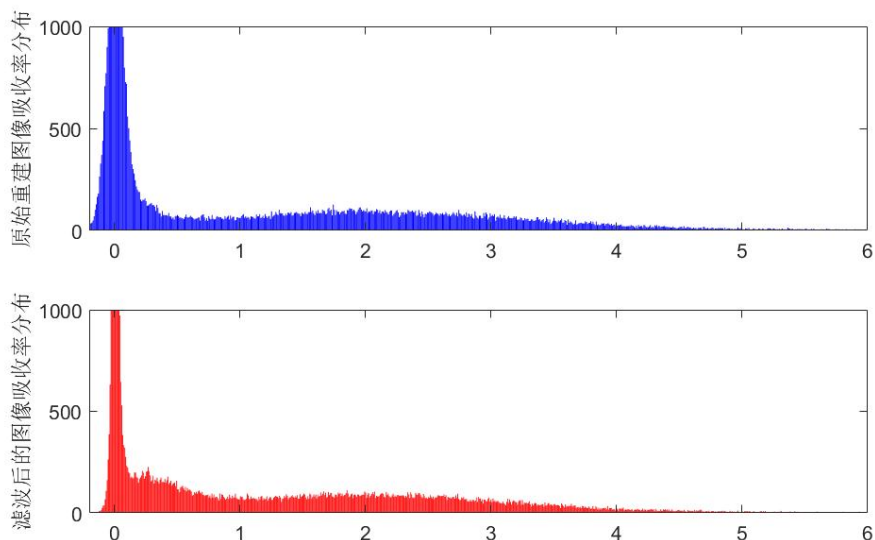


图 23: 问题三投影强度频数直方图（上）与滤波所得结果（下）

4.4 问题四

4.4.1 模型精度与稳定性分析

对于问题一中的标定模型，精度分析如??所述，标定角度的相对误差小于 10^{-4} 。

为分析标定模型稳定性，引入高斯白噪声进行参数标定，将入射角度 θ_i 的预测误差作为稳定性评估指标：

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{180} (\theta_i - \theta_{si})^2} \quad (46)$$

其中下标 s 表示标准值。对于强度不同的高斯白噪声，稳定性分析结果如下：

测次	: $-23.5dB$: $-25dB$: $-26.5dB$
1	2.6355	1.8680	0.5629
2	0.2412	1.9415	0.4724
3	0.3962	4.1245	1.6861
4	1.7213	1.7607	0.1688
5	1.5579	2.6330	1.4477

表 4: 稳定性分析结果

由于整个计算过程比较复杂，从随机过程的角度出发分析引入高斯噪声之后造成的影响有着比较大的困难，但可以利用蒙特卡罗的思想，通过观察结果来描述高斯噪声带来的



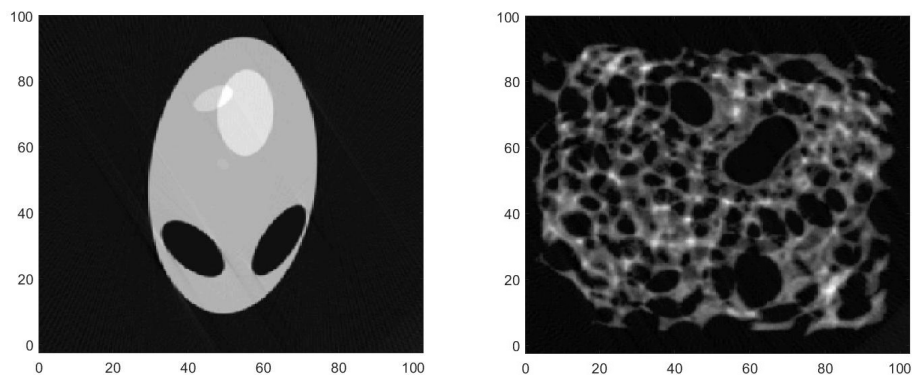


图 24: 未知介质的几何信息

影响。

由表格可以看出，随着噪声的增大，计算得到的角度与无噪声标定角度的偏差逐渐增大，且其具体数值存在较大的波动。观察标定角度的图像（图??），可以发现在绝大部分角度上，标定角度与无噪声标定角度符合得比较好，只有在红圈标出的区间中有着较大的偏差，且这个偏差值并不稳定。根据之前的误差分析，造成这个问题的主要原因是该角度处的误差传递系数很大，略有测量误差就会导致很大的偏差，在这个角度附近难以精确定标。

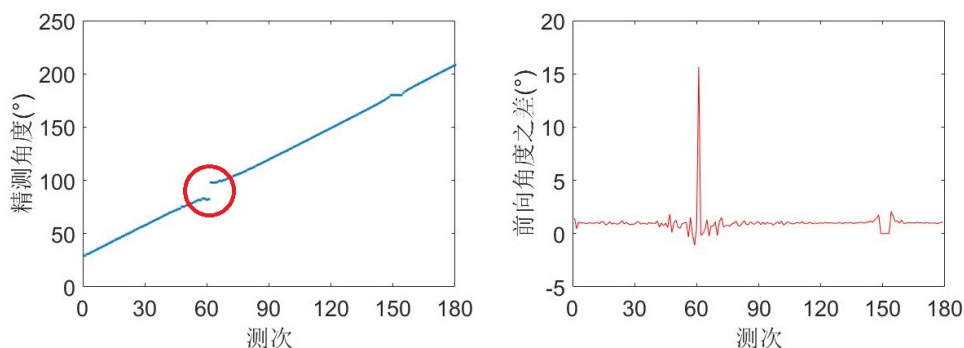


图 25: 稳定性分析结果

4.4.2 新模板设计与标定模型的建立

设计思路

前文分析标定模型精度的问题时曾指出在 $\cos\theta \rightarrow 1$ 或 $\cos\theta \rightarrow 0$ 时，通过求 $\cos^2\theta$ 来求解 θ 会有极大的误差传递系数，解对数据波动十分敏感。此外，我们在检验其稳定性的时候发现，如图??所示，当对原始数据加入高斯噪声之后，求得的标定角度在大部分地方都



是接近于无噪声时的标定值，但是在红圈标注处，与无噪声标定值的差距较为明显，且不同次实验区别很大，这正是该位置的敏感性造成的，甚至会导致出现函数值略大于1（例如 $\arccos(1.0001)$ ）等无法计算的情况，这也正是原标定模型的一个不尽如人意的地方。

为了解决这个问题，我们尝试在原标定模型上进行修改来满足更高精度的需求，依照第一题求解的经验，为了得到更精确的解，应当尽可能多的利用上探测器探测到的各个吸收强度值，即考虑用拟合函数的方法来减小单个强度值的测量误差，这就要求模板的投影函数表达式应足够简单，方便拟合。在这一点上，我们首先考虑了用多边形，如正三角形、正方形等，其拟合函数都是分段直线，十分方便，但是在实践过程中发现，虽然没有 $\arccos(x)$ 的敏感性困难，但是其本身对于形状的精度有很高的要求，如图??所示,正三角形的顶点偏离一个像素，就会对拟合结果有很大影响，这引入了一种新的敏感性，对工艺有极高的要求，故认为该方法不合理。

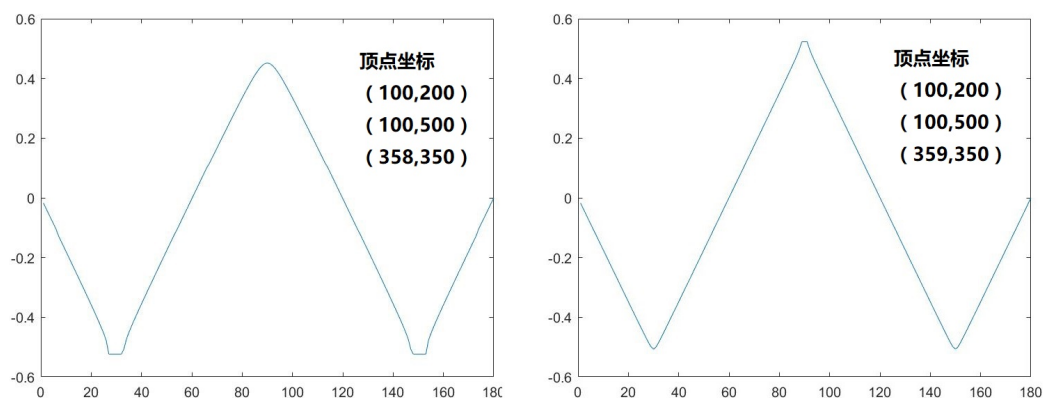


图 26: 正三角模型入射角度拟合曲线

考虑到椭圆投影函数为二次多项式，方便拟合，蕴含信息量足够丰富，最终选择在新模型中保留椭圆形。

解决敏感性的一个方法就是通过两组数据来求得角度值，要求这两组数据在各个位置敏感性不同，于是对于每个位置，可以选出敏感性较低的数据作为测量值，这会对结果有很大的改善。因此，设计新的模板为两个椭圆形介质，如图??所示，其中大椭圆半长轴为40mm，半短轴为15mm，与原模板一致。另外一个椭圆置于原模板中圆的位置附近，长轴与大椭圆长轴45度角，具体参数如图所示。

对该模板进行定性分析，由于两个椭圆的朝向不一样，可以解决单个椭圆定角度时存在敏感角度的问题（只要设计好夹角，使得其各自的敏感角度不重合）。另一方面，由于取消了原模板中的圆，需要用其他方式进行探测器间距 δd 的定标，这对于单个椭圆是比较麻烦的，但是对于两个椭圆，就可以利用其几何关系，经过三角运算得到，尤其在夹角为45°时更为简单。最后由于椭圆大小区别比较明显，提取出两条曲线并没有本质的困难，



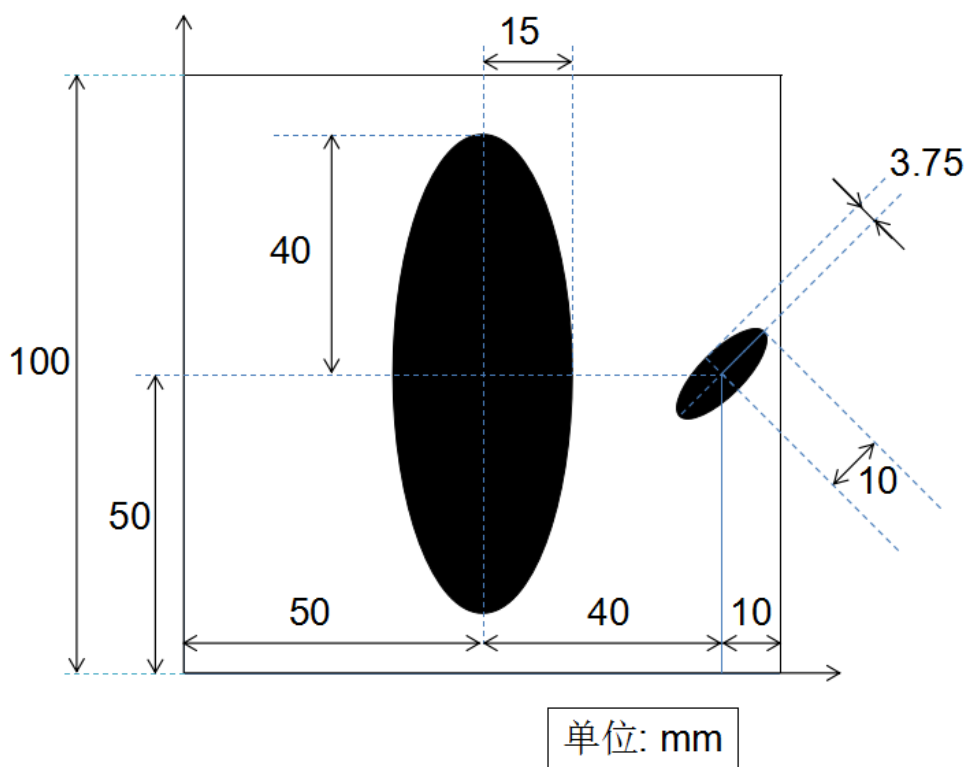


图 27: 新模板设计示意图

与此前的圆形模板算法类似。

标定模型

探测器单元间距 Δ 首先参照原标定模型，将中心椭圆与斜置椭圆的采样数据分离，分别拟合采样函数。

设拟合所得中心椭圆投影强度曲线为：

$$h_1^2(l, \theta) = p_1 l^2 + p_2 l + p_3 \quad (47)$$

根据式??，入射角度与中心椭圆投影曲线存在如下关系：

$$\cos 2\theta = -\frac{4(2a_1^2 - b_1^2)p_1^2 + (4p_3^2 p_1 - p_2^2)\Delta d^2}{8p_1^2(b_1^2 - a_1^2)} \quad (48)$$

设拟合所得斜置椭圆投影强度曲线为：

$$h_3^2(l, \theta) = r_1 l^2 + r_2 l + r_3 \quad (49)$$



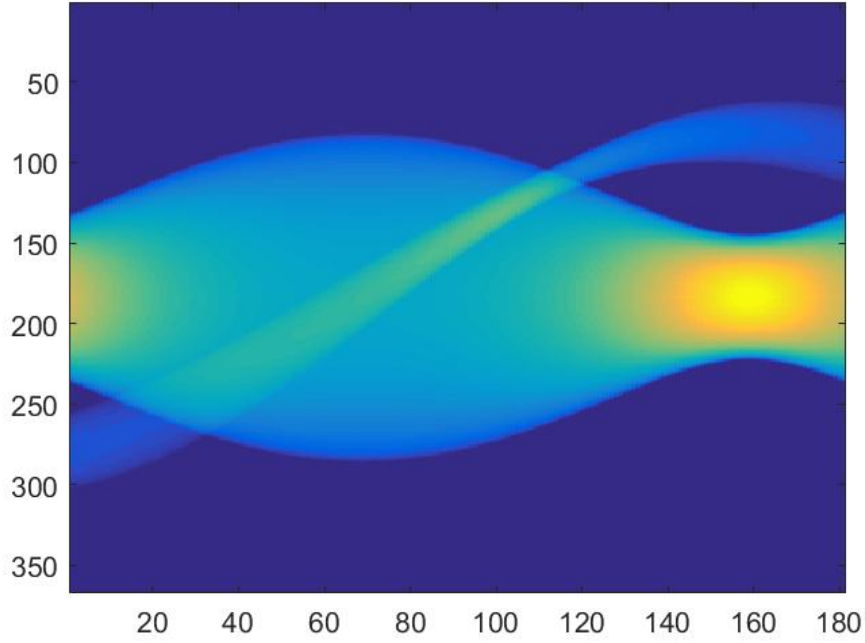


图 28: 双椭圆模型采样数据图

由于中心椭圆与斜置椭圆的长轴夹角为 45° ，对式??将 θ 以 $\theta + 45^\circ$ 代入，得入射角度与斜置椭圆投影曲线的关系为：

$$\cos 2(\theta + 45^\circ) = \sin 2\theta = -\frac{4(2a_2^2 - b_2^2)r_1^2 + (4r_3^2r_1 - r_2^2)\Delta d^2}{8r_1^2(b_2^2 - a_2^2)} \quad (50)$$

为方便表述，记

$$A_1 = -\frac{(4p_3^2p_1 - p_2^2)\Delta d^2}{8p_1^2(b_1^2 - a_1^2)}$$

$$B_1 = -\frac{4(2a_1^2 - b_1^2)p_1^2}{8p_1^2(b_1^2 - a_1^2)}$$

$$A_2 = -\frac{(4r_3^2r_1 - r_2^2)\Delta d^2}{8r_1^2(b_2^2 - a_2^2)}$$

$$B_2 = -\frac{4(2a_2^2 - b_2^2)r_1^2}{8r_1^2(b_2^2 - a_2^2)}$$

则有

$$\cos 2\theta = A_1\Delta d^2 + B_1 \quad (51)$$

$$\sin 2\theta = A_2\Delta d^2 + B_2 \quad (52)$$



联立求解从而确定 Δd 的取值：

$$\Delta d = \frac{-(A_1 B_1 + A_2 B_2) + \sqrt{2A_1 A_2 B_1 B_2 - A_1^2 B_2^2 - A_2^2 B_1^2 + A_1^2 + A_2^2}}{A_1^2 + A_2^2} \quad (53)$$

入射角度 θ 将各次计算的 Δd 取平均，代回求解出 $\cos 2\theta$ 及 $\sin 2\theta$ 。且这两个值是从两套拟合数据中独立计算的，其误差几乎不存在耦合关系。

分析该计算的敏感性，由误差传递公式：

$$\delta(\theta) = \frac{\Delta(\cos 2\theta)}{|-2\sin 2\theta|} \quad (54)$$

$$\delta(\theta) = \frac{\Delta(\sin 2\theta)}{|2\cos 2\theta|} \quad (55)$$

因为 $\sin 2\theta$ 与 $\cos 2\theta$ 不可能同时为0，所以无需担心敏感性极强的问题。

继续分析 $\sin 2\theta$ 与 $\cos 2\theta$ 的误差，以 $\cos 2\theta$ 为例：

$$\delta(\cos 2\theta) = \left| \frac{2b_1}{(b_1^2 - a_1^2)^2} \left(\frac{2C_1 \Delta d}{K_1} + 2a_1^2 \right) \right| \delta(b_1) + \left| \frac{2a_1}{(b_1^2 - a_1^2)^2} \left(\frac{2C_1 \Delta d}{K_1} + 2b_1^2 \right) \right| \delta(a_1) \quad (56)$$

在本题中代入具体数值可知 $\frac{2C_1 \Delta d}{K_1}$ 一般比 a_1^2 小10倍以上，故可以认为误差近似等于：

$$\delta(\cos 2\theta) \approx \left| \frac{4b_1 a_1^2}{(b_1^2 - a_1^2)^2} \right| \delta(b_1) + \left| \frac{4a_1 b_1^2}{(b_1^2 - a_1^2)^2} \right| \delta(a_1) \quad (57)$$

$\sin 2\theta$ 的误差与之同理，只需将对应的尺寸换成小椭圆的尺寸即可。因为小椭圆尺寸是大椭圆的 $\frac{1}{T}$ 倍，从公式中可以看到误差与尺寸的-1次方成正比，所以：

$$\sin 2\theta \approx T \cos 2\theta \quad (58)$$

记由 $\cos 2\theta$ 和 $\sin 2\theta$ 算得的 θ 分别为 θ_c 和 θ_s ，为了尽可能减小的误差，记

$$\theta = u\theta_c + v\theta_s \quad (59)$$

其中 $u + v = 1$ ，则

$$E(\theta^2) = E(u\theta_c + v\theta_s)^2 \quad (60)$$

因为两者的噪声来源基本独立，故认为其协方差系数为0，于是

$$E(\theta^2) = E(u^2\theta_c^2 + v^2\theta_s^2) \quad (61)$$

对上式求导，可知 $u = \frac{\theta_s^2}{\theta_c^2 + \theta_s^2}$ 时，误差的方差达到最小。可以求得：

$$E(\theta_s^2) \approx \frac{E(\delta(\cos 2\theta)^2)}{4\sin^2 2\theta_c} \quad (62)$$



$$E(\theta_c^2) \approx \frac{T^2 E(\delta(\sin 2\theta)^2)}{4 \cos^2 2\theta_s} \quad (63)$$

所以，此时：

$$u = \frac{T \sin^2 2\theta_c}{\cos^2 2\theta_s + T \sin^2 2\theta_c} \quad (64)$$

$$v = \frac{\cos^2 2\theta_s}{\cos^2 2\theta_s + T \sin^2 2\theta_c} \quad (65)$$

从而求解 $\theta = u\theta_c + v\theta_s$ 的值。

旋转中心位置(x, y) 利用分别拟合的两条采样函数曲线，经由变量替换得到中心椭圆与斜置椭圆的投影强度曲线，从而计算两椭圆中心 O_1, O_2 的投影位置曲线。以 O_1, O_2 与旋转中心 O 为顶点构造三角形，由 O_1, O_2 的投影位置曲线可确定三角形的内角，从而在正方形托盘上定位出旋转中心的位置。此部分计算与原模型基本相同，此处不再赘述。

5 模型评价

本文以Radon变换等物理背景为理论依据，进行了二维平面上的平行射线CT成像系统的建模与分析。

在由题目所给模板对CT系统进行标定时，先使用图像处理的形态学操作得到了椭圆投影的宽度变化规律，计算得到各角度参数、旋转中心的估计值，由此进一步处理图像，将椭圆和圆的两部分影响独立出来并分别进行拟合，在题述条件下，由拟合结果得到了高精度的结果。

由于原标定模型的单个椭圆的定标能力限制，该模型在敏感角度附近难以达到理想精度，在引入高斯白噪声的情况下产生了较大的测量误差。

因此原标定模型对噪声的抵抗能力较差，应用范围受限。但在经过良好滤波除噪的情况下，能达到较为理想的标定精度。

新模型针对原有标定模型的固有缺陷进行了改进，以斜置椭圆替代圆形模板，在大致保留原有模型特性的基础上，引入了相位不同的新标定曲线，通过双椭圆标定曲线的互补，克服了原有的敏感性缺陷，在同等条件下，与原模型相比可以取得更高的标定精度。

参考文献

- [1] 庄天戈. CT原理与算法[M]. 上海交通大学出版社, 1992.
- [2] 沈建仪. 椭圆平行切线的性质及应用[J]. 中学数学教学, 1987(04).



- [3] 郭立倩. CT系统标定与有限角度CT重建方法的研究[D]. 大连理工大学, 2016.
- [4] Turbell H. Cone-beam reconstruction using filtered backprojection /[J]. Linköping University, 2001.
- [5] JiangHsieh, Hsieh, 张朝宗,等. 计算机断层成像技术:原理、设计、伪像和进展[J]. 2006.
- [6] 高上凯. 医学成像系统.第2版[M]. 清华大学出版社, 2010.



附录

```
"problem_1.m"
clc;close all;
%% 获取数据
dataFile = '..\A题附件.xls';
data = xlsread(dataFile,2);

%% 参数设定
lenTheta = 180;          % 总测量角度数
pixScale = 512;          % 单次测量像素数目
normScale= 100;          % 标准尺度(100mm)
graphScale=256;          % 目标像素尺度(256pixel)
a=40; b=15; R=4;         % 图形参数
normEllips0 = [50,50]; % 椭圆中心位置
normCircle0 = [95,50]; % 圆形中心位置

%% 基本信息提取
figure('Name','各次测量的接收值总和');plot(sum(data),'b. ');hold on;
axis([1,lenTheta,min(sum(data))*0.997,max(sum(data))*1.003]);
plot([1,lenTheta],[mean(sum(data)),mean(sum(data))],'r-');
text(lenTheta-50,max(sum(data))*1.003-20,['std=',num2str(std(sum(data)))], 'FontSize',14);
legend('各次测量的接收值求和','多次测量的接收值均值');

%% 粗测尺度及角度
% 获取曲线边缘,得到椭圆宽度变化
% figure('Name','原始数据灰度图');colormap(gray);imagesc(data);
curv = abs(diff(data));
curv = curv > 1;
se = strel('square',3);
curvEllips = imopen(curv,se);
curvEllips = bwareaopen(curvEllips,500);

% 得到每一角度对应的椭圆宽度数据
widthEllips = zeros(1,lenTheta);
```




```

for j=1:lenTheta
    widthEllips(j) = find(curvEllips(:,j),1,'last')-(find(curvEllips(:,j),1,'first'));
end

% 粗略计算角度和尺度
roughScale = max(widthEllips)/2/a;
normWidthEllipse = widthEllips/roughScale;
roughTheta = zeros(1,lenTheta);
for j=1:lenTheta
    roughTheta(j) = 180/pi*real(acot(sqrt((4*a^2-normWidthEllipse(j)^2)/(normWidthEllips
end

% 拟合得到各个角度,粗略得到逆变换图
diffRoughTheta = diff(roughTheta);
breakP1 = find(diffRoughTheta<-0.1,1);
breakP2 = find(diffRoughTheta(breakP1+10:end)>0.1,1)+breakP1+10;
roughTheta(1:breakP1)=roughTheta(1:breakP1);
roughTheta(breakP1+1:breakP2-1)=180-roughTheta(breakP1+1:breakP2-1);
roughTheta(breakP2:end)=180+roughTheta(breakP2:end);
[temp,~]=polyfit(1:lenTheta,roughTheta,1);
k_temp = temp(1);
b_temp = temp(2);
linerTheta = (1:lenTheta)*k_temp+b_temp;
figure('Name','粗测角度及拟合情况');plot(1:lenTheta,roughTheta,'r. ');
hold on;plot(1:lenTheta,linerTheta,'b-');
legend('粗测的角度','拟合的角度');
roughMap = iradon(data,linerTheta,pixScale);

% 旋转修正
Status = regionprops((roughMap>max(max(roughMap))/5));
[~,orderEllips] = max([Status(:).Area]);
[~,orderCircle] = min([Status(:).Area]);
pixEllips0 = Status(orderEllips).Centroid;
pixCircle0 = Status(orderCircle).Centroid;
fixAng = angle((pixCircle0-pixEllips0)*[1;1i]);

```



```

linerTheta = linerTheta+fixAng*180/pi;
roughMap = iradon(data,linerTheta,pixScale);

% 得到小圆圆心在图像中位置
Status = regionprops((roughMap>max(max(roughMap))/5));
[~,orderCircle] = min([Status(:).Area]);
pixCircle0 = Status(orderCircle).Centroid;
pix0 = [pixScale/2,pixScale/2];

% 得到轴在探测器阵列中的投影位置
[~,minWidthEllipsPos] = min(widthEllips);
tempStart = find(data(:,minWidthEllipsPos)>0,1,'first');
tempEnd = find(data(tempStart:end,minWidthEllipsPos)==0,1,'first')+tempStart;
AxisPosInSensors = (tempStart+tempEnd)/2+abs(pix0(1)-pixCircle0(1));

%% 拟合得到精确尺寸和角度
% 拟合准备
pixCircleCenter = zeros(1,lenTheta);
pixEllipsCenter = zeros(1,lenTheta);
fitEllipsData= data;
fitCircleData= data;
fitEllipsAns = zeros(pixScale,lenTheta);
fitCircleAns = zeros(pixScale,lenTheta);
fitAns = zeros(3,lenTheta);
fitMin = max(max(data))*0.01;
x = 1:pixScale;

% 拟合过程
for j=1:lenTheta
    % 椭圆的宽度曲线拟合
    pixCircleCenter(j) = int16(AxisPosInSensors - norm(pix0-pixCircle0)*cos(linerTheta(j)));
    fitEllipsData(int16(pixCircleCenter(j)-2*R*roughScale:pixCircleCenter(j)+2*R*roughScale),j) = data(int16(pixCircleCenter(j)-2*R*roughScale:pixCircleCenter(j)+2*R*roughScale),j);
    fitEllipsRange = find(fitEllipsData(:,j)>fitMin);
    fitEllips = polyfit(x(fitEllipsRange)',fitEllipsData(fitEllipsRange,j).^2,2);
    fitEllipsAns(:,j) = (fitEllips(1)*x.^2+fitEllips(2)*x+fitEllips(3))';
end

```



```

fitEllipsAns(fitEllipsAns(:,j)<0,j)=0;
fitEllipsAns(:,j) = sqrt(fitEllipsAns(:,j));

% 圆的宽度曲线拟合
fitCircleData(:,j) = data(:,j)-fitEllipsAns(:,j);
fitCircleRange = find(fitCircleData(:,j)>fitMin);
fitCircle = polyfit(x(fitCircleRange)',fitCircleData(fitCircleRange,j).^2,2);
fitCircleAns(:,j) = (fitCircle(1)*x.^2+fitCircle(2)*x+fitCircle(3))';
fitCircleAns(fitCircleAns(:,j)<0,j)=0;
fitCircleAns(:,j) = sqrt(fitCircleAns(:,j));

% 拟合结果处理
pixCircleCenter(j) = -fitCircle(2)/fitCircle(1)/2;
pixEllipsCenter(j) = -fitEllips(2)/fitEllips(1)/2;
deltaL = 8/(sqrt(fitCircle(2)^2-4*fitCircle(1)*fitCircle(3))/abs(fitCircle(1)));
G = -8/((fitCircle(2)^2-4*fitCircle(1)*fitCircle(3))/fitCircle(1)/4);
temp = -deltaL^2*(fitEllips(3)-fitEllips(2)^2/fitEllips(1)/4)/fitEllips(1);
cosTheta2 = (temp/b^2-a^2/b^2)/(1-a^2/b^2);
fitAns(:,j) = [deltaL,G,cosTheta2];
end

figure('Name','椭圆部分数据图');colormap(gray);imagesc(fitEllipsData);
figure('Name','椭圆部分拟合图');colormap(gray);imagesc(fitEllipsAns);
figure('Name','小圆部分数据图');colormap(gray);imagesc(fitCircleData);
figure('Name','小圆部分拟合图');colormap(gray);imagesc(fitCircleAns);
figure('Name','拟合结果与实际数据对比图');
temp = 150;
subplot(2,1,1);hold on;
plot(fitEllipsData(:,temp),'b. ');plot(fitEllipsAns(:,temp),'g-');
plot(fitCircleData(:,temp),'b. ');plot(fitCircleAns(:,temp),'g-');
subplot(2,1,2);hold on;
plot(fitEllipsAns(:,temp)-fitEllipsData(:,temp),'r-');
plot(fitCircleAns(:,temp)-fitCircleData(:,temp),'r-');

% 得到拟合出的角度并修正方向
fixTheta = 1/2*acos(2.*fitAns(3,:)-1)*180/pi;

```



```

diffFixTheta = diff(fixTheta);
breakP1 = find(diffFixTheta<0,1);
breakP2 = find(diffFixTheta(breakP1+1:end)>0,1)+breakP1+1;
fixTheta(1:breakP1)=fixTheta(1:breakP1);
fixTheta(breakP1+1:breakP2-1)=180-fixTheta(breakP1+1:breakP2-1);
fixTheta(breakP2:end)=180+fixTheta(breakP2:end);
figure('Name','修正后的各个角度');subplot(1,2,1);plot(fixTheta,'.');
subplot(1,2,2);plot(diff(fixTheta),'r-');
set(gcf,'Position',[500 500 900 400]);

%% 相关参数的求解
% 得到深度系数和探测器间距
G = mean(fitAns(2,:));
deltaL = mean(fitAns(1,:));
[interpData,intepTheta]=interpForIradon(fixTheta,data);
fixMap = iradon(interpData,intepTheta,'spline','Hann',pixScale);

% 精测旋转中心
Status = regionprops((fixMap>max(max(fixMap))/5));
[~,orderEllips] = max([Status(:).Area]);
[~,orderCircle] = min([Status(:).Area]);
pixEllips0 = Status(orderEllips).Centroid;
pixCircle0 = Status(orderCircle).Centroid;
pix0 = [pixScale/2,pixScale/2];
norm0 = ((2*pix0-pixEllips0-pixCircle0)*deltaL+normEllips0+normCircle0)/2;
figure('Name','修正的重建图像');colormap(gray);imshow(fixMap);
hold on;plot(pix0(1),pix0(2),'ro');
plot([pixEllips0(1),pixCircle0(1)],[pixEllips0(2),pixCircle0(2)],'ko');
plot([pix0(1),pixEllips0(1),pixCircle0(1),pix0(1)],[pix0(2),pixEllips0(2),pixCircle0(2),pix0(2)],'k-');hold on;

% 绘制原尺寸图内的旋转中心
figure('Name','旋转中心示意图');
plot([0,100,100,0],[0,0,100,100],'k-');hold on;
tempLen = 1000;
xEllipsPos = zeros(tempLen);

```



```

yEllipsPos = zeros(tempLen);
xCirclePos = zeros(tempLen);
yCirclePos = zeros(tempLen);
for i=1:tempLen
    sita = i/tempLen*2*pi;
    xEllipsPos(i) = b*cos(sita)+normEllips0(1);
    yEllipsPos(i) = a*sin(sita)+normEllips0(2);
    xCirclePos(i) = R*cos(sita)+normCircle0(1);
    yCirclePos(i) = R*sin(sita)+normCircle0(2);
end
plot(xEllipsPos,yEllipsPos,'k-');
plot(xCirclePos,yCirclePos,'k-');
plot(norm0(1),norm0(2),'ro');
plot([normEllips0(1),normCircle0(1)], [normEllips0(2),normCircle0(2)], 'ko');
plot([norm0(1),normEllips0(1),normCircle0(1),norm0(1)], [norm0(2),normEllips0(2),normCircle0(2),norm0(2)], 'k');
axis('equal');
set(gca,'YDir','reverse');

% 求解'吸收率'系数
figure('Name','重建图形的吸收率分布');hist(fixMap(:),1000);
[fixKm,fixKmC] = kmeans(fixMap(:),2);
Gt = max(fixKmC)/1.00;

%% 保存数据
CalibrationResults = {
    {'Theta'; roundn(fixTheta,-4)};
    {'norm0'; roundn(norm0, -4)};
    {'deltaL'; roundn(deltaL, -4)};
    {'G'; roundn(G, -4)};
    {'Gt'; roundn(Gt, -4)}
};
for i=1:length(CalibrationResults)
    xlswrite('Calibration.xls',{CalibrationResults{i}{1}},1,['A' num2str(i)]);
    xlswrite('Calibration.xls', CalibrationResults{i}{2} ,1,['B' num2str(i)]);
end

```



```

"problem_2.m"
clc;close all;
%% 获取数据
dataFile = '..\A题附件.xls';
data = xlsread(dataFile,3);
points=xlsread(dataFile,4);
CalibrationFile = 'Calibration.xls';
CalibrationData = xlsread(CalibrationFile,1);
Theta = CalibrationData(1,:);
norm0 = CalibrationData(2,1:2);
deltaL= CalibrationData(3,1);
G      = CalibrationData(4,1);
Gt     = CalibrationData(5,1);

%% 计算/显示/保存
pixScale = 512;
desScale = 256;
[interpData,interpTheta] = interpForIradon(Theta,data);
Map = iradon(interpData,interpTheta,'spline','Hann',pixScale)./Gt;
figure('Name','直接逆Radon变换');colormap(gray);imagesc(Map);

% 图像插值
pix0 = [pixScale/2,pixScale/2];
desGraph = zeros(desScale);
for i=1:desScale
    for j=1:desScale
        pixPoint = pix0+([i-0.5,j-0.5]./desScale*100-[norm0(2),norm0(1)])/deltaL*pixScale;
        desGraph(i,j) = interpAbsorbMap(pixPoint,Map);
    end
end
figure('Name','吸收率平面分布图');colormap(gray);imagesc(desGraph);

```



```

% 图像wiener滤波
K=wiener2(desGraph,[5 5]);
figure('Name','吸收率占比分布图');
[tempA,tempB]=hist(desGraph(:),1000);
subplot(211);bar(tempB,tempA,'b');axis([-0.2,1.6,0,4000]);
ylabel('原始重建图像吸收率分布');
[tempA,tempB]=hist(K(:),1000);
subplot(212);bar(tempB,tempA,'r');axis([-0.2,1.6,0,4000]);
ylabel('滤波后的图像吸收率分布');

figure('Name','滤波前后对比图');
subplot(121);imagesc(desGraph); ylabel('原始图像');
subplot(122);imagesc(K); ylabel('滤波处理');
set(gcf,'Position',[500 500 900 400]);

% 计算各点吸收率
Absorbtion = zeros(1,length(points));
fprintf('各点吸收率数据: \n');
pixPoints = zeros(length(points),2);
for i=1:length(points)
    pixPoints(i,:) = [points(i,1),100-points(i,2)]/100*desScale*pixScale/512;
    Absorbtion(i) = interpAbsorbMap([pixPoints(i,2),pixPoints(i,1)],K);
    fprintf('%.4f\n',Absorbtion(i));
end

% 显示各点位置
figure('Name','各点分布图');colormap(gray);imagesc(K);hold on;
for i=1:length(points)
    plot(points(i,1)*desScale/100,(100-points(i,2))*desScale/100,'r*');
    text(points(i,1)*desScale/100,(100-points(i,2))*desScale/100,num2str(roundn(Absorbti
end

% 保存数据
xlswrite('problem2.xls',roundn(K,-4));

```



```

"problem_3.m"
clc;close all;
%% 获取数据
dataFile = '..\A题附件.xls';
data = xlsread(dataFile,5);
points=xlsread(dataFile,4);
CalibrationFile = 'Calibration.xls';
CalibrationData = xlsread(CalibrationFile,1);
Theta = CalibrationData(1,:);
norm0 = CalibrationData(2,1:2);
deltaL= CalibrationData(3,1);
G      = CalibrationData(4,1);
Gt     = CalibrationData(5,1);

%% 计算/显示/保存
pixScale = 512;
desScale = 256;
[interpData,interpTheta] = interpForIradon(Theta,data);
Map = iradon(interpData,interpTheta,'spline','Hann',pixScale)./Gt;
figure('Name','直接逆Radon变换');colormap(gray);imagesc(Map);

% 图像插值
pix0 = [pixScale/2,pixScale/2];
desGraph = zeros(desScale);
for i=1:desScale
    for j=1:desScale
        pixPoint = pix0+([i-0.5,j-0.5]./desScale*100-[norm0(2),norm0(1)])/deltaL*pixScale;
        desGraph(i,j) = interpAbsorbMap(pixPoint,Map);
    end
end
figure('Name','吸收率平面分布图');colormap(gray);imagesc(desGraph);

% 图像wiener滤波

```




```

K=wiener2(desGraph,[5 5]);
figure('Name','吸收率占比分布图');
[tempA,tempB]=hist(desGraph(:),1000);
subplot(211);bar(tempB,tempA,'b');axis([-0.2,6,0,1000]);
ylabel('原始重建图像吸收率分布');
[tempA,tempB]=hist(K(:),1000);
subplot(212);bar(tempB,tempA,'r');axis([-0.2,6,0,1000]);
ylabel('滤波后的图像吸收率分布');

figure('Name','滤波前后对比图');
subplot(121);imagesc(desGraph); ylabel('原始图像');
subplot(122);imagesc(K); ylabel('滤波处理');
set(gcf,'Position',[500 500 900 400]);

% 计算各点吸收率
Absorbtion = zeros(1,length(points));
fprintf('各点吸收率数据: \n');
pixPoints = zeros(length(points),2);
for i=1:length(points)
    pixPoints(i,:) = [points(i,1),100-points(i,2)]/100*desScale*pixScale/512;
    Absorbtion(i) = interpAbsorbMap([pixPoints(i,2),pixPoints(i,1)],desGraph);
    fprintf('%.4f\n',Absorbtion(i));
end

% 显示各点位置
figure('Name','各点分布图');colormap(gray);imagesc(desGraph);hold on;
for i=1:length(points)
    plot(points(i,1)*desScale/100,(100-points(i,2))*desScale/100,'r*');
    text(points(i,1)*desScale/100,(100-points(i,2))*desScale/100,num2str(roundn(Absorbti
end

% 保存数据
xlswrite('problem3.xls',roundn(desGraph,-4));

```



```

"problem_4.m"
clc;close all;
%% 获取数据
dataFile = '..\A题附件.xls';
originData = xlsread(dataFile,2);
CalibrationFile = 'Calibration.xls';
CalibrationData = xlsread(CalibrationFile,1);
Theta = CalibrationData(1,:);
norm0 = CalibrationData(2,1:2);
deltaLs= CalibrationData(3,1);
G      = CalibrationData(4,1);
Gts    = CalibrationData(5,1);

%% 参数设定
lenTheta = 180;          % 总测量角度数
pixScale = 512;          % 单次测量像素数目
normScale= 100;          % 标准尺度(100mm)
graphScale=256;          % 目标像素尺度(256pixel)
a=40; b=15; R=4;         % 图形参数
normEllips0 = [50,50]; % 椭圆中心位置
normCircle0 = [95,50]; % 圆形中心位置

%% 基本信息提取
data = originData;

%% 粗测尺度及角度
% 获取曲线边缘，得到椭圆宽度变化
% figure('Name','原始数据灰度图');colormap(gray);imagesc(data);
curv = abs(diff(data));
curv = curv > 1;
se = strel('square',3);
curvEllips = imopen(curv,se);
curvEllips = bwareaopen(curvEllips,500);

```



```

% 得到每一角度对应的椭圆宽度数据
widthEllips = zeros(1,lenTheta);
for j=1:lenTheta
    widthEllips(j) = find(curvEllips(:,j),1,'last')-(find(curvEllips(:,j),1,'first'));
end

% 粗略计算角度和尺度
roughScale = max(widthEllips)/2/a;
normWidthEllipse = widthEllips/roughScale;
roughTheta = zeros(1,lenTheta);
for j=1:lenTheta
    roughTheta(j) = 180/pi*real(acot(sqrt((4*a^2-normWidthEllipse(j)^2)/(normWidthEllipse(j)
end

% 拟合得到各个角度,粗略得到逆变换图
diffRoughTheta = diff(roughTheta);
breakP1 = find(diffRoughTheta<-0.1,1);
breakP2 = find(diffRoughTheta(breakP1+10:end)>0.1,1)+breakP1+10;
roughTheta(1:breakP1)=roughTheta(1:breakP1);
roughTheta(breakP1+1:breakP2-1)=180-roughTheta(breakP1+1:breakP2-1);
roughTheta(breakP2:end)=180+roughTheta(breakP2:end);
[temp,~]=polyfit(1:lenTheta,roughTheta,1);
k_temp = temp(1);
b_temp = temp(2);
linerTheta = (1:lenTheta)*k_temp+b_temp;
roughMap = iradon(data,linerTheta,pixScale);

% 旋转修正
Status = regionprops((roughMap>max(max(roughMap))/5));
[~,orderEllips] = max([Status(:).Area]);
[~,orderCircle] = min([Status(:).Area]);
pixEllips0 = Status(orderEllips).Centroid;
pixCircle0 = Status(orderCircle).Centroid;
fixAng = angle((pixCircle0-pixEllips0)*[1;1i]);
linerTheta = linerTheta+fixAng*180/pi;

```



```

roughMap = iradon(data,linerTheta,pixScale);

% 得到小圆圆心在图像中位置
Status = regionprops((roughMap>max(max(roughMap))/5));
[~,orderCircle] = min([Status(:).Area]);
pixCircle0 = Status(orderCircle).Centroid;
pix0 = [pixScale/2,pixScale/2];

% 得到轴在探测器阵列中的投影位置
[~,minWidthEllipsPos] = min(widthEllips);
tempStart = find(data(:,minWidthEllipsPos)>0,1,'first');
tempEnd = find(data(tempStart:end,minWidthEllipsPos)==0,1,'first')+tempStart;
AxisPosInSensors = (tempStart+tempEnd)/2+abs(pix0(1)-pixCircle0(1));

%% 加入高斯噪声，分析精测部分的敏感性问题
gaussPower = 0.00002;
data = imnoise(originData./max(max(originData)),'gaussian',0,gaussPower).*max(max(originData),1);

%% 拟合得到精确尺寸和角度
% 拟合准备
pixCircleCenter = zeros(1,lenTheta);
pixEllipsCenter = zeros(1,lenTheta);
fitEllipsData= data;
fitCircleData= data;
fitEllipsAns = zeros(pixScale,lenTheta);
fitCircleAns = zeros(pixScale,lenTheta);
fitAns = zeros(3,lenTheta);
fitMin = max(max(data))*0.03;
x = 1:pixScale;

% 拟合过程
for j=1:lenTheta
    % 椭圆的宽度曲线拟合
    pixCircleCenter(j) = int16(AxisPosInSensors - norm(pix0-pixCircle0)*cos(linerTheta(j)));
    fitEllipsData(int16(pixCircleCenter(j))-2*R*roughScale:pixCircleCenter(j)+2*R*roughScale) = data(int16(pixCircleCenter(j))-2*R*roughScale:pixCircleCenter(j)+2*R*roughScale);
end

```



```

fitEllipsRange = find(fitEllipsData(:,j)>fitMin);
fitEllips = polyfit(x(fitEllipsRange)',fitEllipsData(fitEllipsRange,j).^2,2);
fitEllipsAns(:,j) = (fitEllips(1)*x.^2+fitEllips(2)*x+fitEllips(3))';
fitEllipsAns(fitEllipsAns(:,j)<0,j)=0;
fitEllipsAns(:,j) = sqrt(fitEllipsAns(:,j));

% 圆的宽度曲线拟合
fitCircleData(:,j) = data(:,j)-fitEllipsAns(:,j);
fitCircleRange = find(fitCircleData(:,j)>fitMin);
fitCircle = polyfit(x(fitCircleRange)',fitCircleData(fitCircleRange,j).^2,2);
fitCircleAns(:,j) = (fitCircle(1)*x.^2+fitCircle(2)*x+fitCircle(3))';
fitCircleAns(fitCircleAns(:,j)<0,j)=0;
fitCircleAns(:,j) = sqrt(fitCircleAns(:,j));

% 拟合结果处理
pixCircleCenter(j) = -fitCircle(2)/fitCircle(1)/2;
pixEllipsCenter(j) = -fitEllips(2)/fitEllips(1)/2;
deltaL = 8/(sqrt(fitCircle(2)^2-4*fitCircle(1)*fitCircle(3))/abs(fitCircle(1)));
G = -8/((fitCircle(2)^2-4*fitCircle(1)*fitCircle(3))/fitCircle(1)/4);
temp = -(fitEllips(3)-fitEllips(2)^2/fitEllips(1)/4)/fitEllips(1);
cosTheta2 = (temp/b^2-a^2/b^2)/(1-a^2/b^2);
fitAns(:,j) = [deltaL,G,temp];
end

figure('Name','椭圆部分拟合图');colormap(gray);imagesc(fitEllipsAns);
figure('Name','小圆部分拟合图');colormap(gray);imagesc(fitCircleAns);

% 得到拟合出的角度并修正方向
G = mean(fitAns(2,:));
deltaL = mean(fitAns(1,:));
temp = deltaL^2.*fitAns(3,:);
temp = (temp./b^2-a^2/b^2)./(1-a^2/b^2);
fixTheta = real(1/2*acos(2.*temp-1)*180/pi);
diffFixTheta = diff(fixTheta);
breakP1 = 61;

```



```

breakP2 = 152;
fixTheta(1:breakP1)=fixTheta(1:breakP1);
fixTheta(breakP1+1:breakP2-1)=180-fixTheta(breakP1+1:breakP2-1);
fixTheta(breakP2:end)=180+fixTheta(breakP2:end);
figure('Name','修正后的各个角度');subplot(1,2,1);plot(fixTheta,'.');
subplot(1,2,2);plot(diff(fixTheta),'r-');

%% 相关参数的求解
[interpData,interpTheta]=interpForIradon(fixTheta,data);
fixMap = iradon(interpData,interpTheta,'spline','Hann',pixScale);
figure('Name','重建图像');colormap(gray);imagesc(fixMap);

% 求解'吸收率'系数
[fixKm,fixKmC] = kmeans(fixMap(:),2);
Gt = max(fixKmC)/1.00;

fprintf('高斯误差强度:%.6f\n',gaussPower);
fprintf('求得角度误差:%.4f\n',std(fixTheta-Theta));

```

"interpAbsorbMap.m"

%% 图形插值

% 输入:带权图和目标坐标点

% 输出:目标坐标点附近点的线性插值得到的权值

```

function val = interpAbsorbMap(Point,Map)
    floorX = floor(Point(1));ceilX = ceil(Point(1));
    floorY = floor(Point(2));ceilY = ceil(Point(2));
    if floorX<1
        floorX = 1;ceilX = floorX+1;
    end
    if ceilX>length(Map)
        ceilX = length(Map);floorX = ceilX-1;
    end
    if floorY<1

```



```

        floorY = 1;ceilY = floorY+1;
    end
    if ceilY>length(Map)
        ceilY = length(Map);floorY = ceilY-1;
    end
    tempY1 = (Map(ceilX,floorY)-Map(floorX,floorY))*(ceilX-floorX)+Map(floorX,floorY);
    tempY2 = (Map(ceilX,ceilY)-Map(floorX,ceilY))*(ceilX-floorX)+Map(floorX,ceilY);
    val = (tempY2-tempY1)*(ceilY-floorY)+tempY1;
end

```

"interpForIradon.m"

```

function [interpData,interpTheta]=interpForIradon(Theta,data)
    interpTheta = Theta(1):(Theta(end)-Theta(1))/(length(Theta)-1):Theta(end);
    interpData = data;
    for j=2:length(Theta)-1
        if interpTheta(j)<Theta(j)
            start = j-1; finish = j;
        else
            start = j;    finish = j+1;
        end
        interpData(:,j) = (data(:,finish)-data(:,start))*(interpTheta(j)-Theta(start))/(
    end
end

```

