附录：

1. figure3.m

%椭圆和圆的函数表达式

x1 = -15:0.01:15;

y1\_p = (40^2-64/9\*x1.^2).^(1/2);

y1\_n = -(40^2-64/9\*x1.^2).^(1/2);

x2 = 41:0.01:49;

y2\_p = (4^2-(x2-45).^2).^(1/2);

y2\_n = -(4^2-(x2-45).^2).^(1/2);

%椭圆和圆作图

plot(-50:0.1:50,zeros(1,1001),'k-'

hold on

plot(zeros(1001,1),-50:0.1:50,'k-')

hold on

plot(x1,y1\_p,'b-')

xlim([-50 50])

ylim([-50 50])

xticks([-50 -40 -30 -20 -10 0 10 20 30 40 50])

xlabel('X')

ylabel('Y')

axis square

hold on

plot(x1,y1\_n,'b-')

hold on

plot(x2,y2\_p,'b-')

hold on

plot(x2,y2\_n,'b-')

hold on

1. Qiexian.m

%line1

a1 = -1.09218;

b1 = 43.2249;

Lx1 = -30:0.01:50;

Ly1 = a1\*Lx1+b1;

plot(Lx1,Ly1,'k-','LineWidth',0.5)

hold on

%line2

b2 = -43.2249;

a2 = 1.09218;

Lx2 = -30:0.01:50;

Ly2 = a2\*Lx2+b2;

plot(Lx2,Ly2,'k-','LineWidth',0.5)

hold on

%椭圆和圆的函数表达式

x1 = -15:0.01:15;

y1\_p = (40^2-64/9\*x1.^2).^(1/2);

y1\_n = -(40^2-64/9\*x1.^2).^(1/2);

x2 = 41:0.01:49;

y2\_p = (4^2-(x2-45).^2).^(1/2);

y2\_n = -(4^2-(x2-45).^2).^(1/2);

%椭圆和圆作图

plot(-50:0.1:50,zeros(1,1001),'k-')

hold on

plot(zeros(1001,1),-50:0.1:50,'k-')

hold on

plot(x1,y1\_p,'b-')

xlim([-50 50])

ylim([-50 50])

xticks([-50 -40 -30 -20 -10 0 10 20 30 40 50])

xlabel('X')

ylabel('Y')

axis square

hold on

plot(x1,y1\_n,'b-')

hold on

plot(x2,y2\_p,'b-')

hold on

plot(x2,y2\_n,'b-')

%算L1L2夹角

alpha = (atan(-a1)+atan(a2))/(2\*pi)\*360;

theta0 = alpha/(109-14);

beta1 = atan(a2)/(2\*pi)\*360-13\*theta0;

%算探测器单元间距

sigma=8/28;

1. Rotation\_center.m

%探测器单元间距

d = 8/28;

%各直线的表达式

%line1: y=a1\*x+b1

%line2: y=a2\*x+b2

%line3: y=a1\*x+b3

%line4: y=a2\*x+b4

%line11: y=a3\*x+b11

%line12: y=a3\*x+b12

%line21: y=a4\*x+b21

%line22: y=a4\*x+b22

%line1 line3

a1 = -1.09218;

b1 = 43.2249;

b3 = b1 - 8/cos(atan(a1));

%line2 line4

b2 = -43.2249;

a2 = 1.09218;

b4 = b2 - 8/cos(atan(a2));

%line1两条垂线

syms a3 b11 b

a3 = -1/a1;

b = solve('(0-a3\*45-b11)^2 = 8^2\*(a3^2+b11^2)', 'b11');

b11 = eval(b);

%line2两条垂线

syms a4 b21 b

a4 = -1/a2;

b = solve('(0-a4\*45-b21)^2 = 8^2\*(a4^2+b21^2)', 'b21');

b21 = eval(b);

%求A,B,C,D坐标（line1,line3,line1两条垂线的交点）

%line1和垂线交点

syms xa xb b11\_1 b11\_2 x

b11\_1 = b11(1);

b11\_2 = b11(2);

x = solve('a1\*xa+b1 = a3\*xa+b11\_1', 'xa');

xa = eval(x);

ya = a1\*xa + b1; % A(xa, ya)

x = solve('a1\*xb+b1 = a3\*xb + b11\_2', 'xb');

xb = eval(x);

yb = a1\*xb + b1; % B(xb, yb)

%line3和垂线交点

syms xc xd x

x = solve('a1\*xc+b3 = a3\*xc+b11\_2', 'xc');

xc = eval(x);

yc = a1\*xc + b3; % C(xc, yc)

x = solve('a1\*xd+b3 = a3\*xd+b11\_1', 'xd');

xd = eval(x);

yd = a1\*xd + b3; % D(xd, yd)

%line2平移后直线line\_2

b22 = b2 + (399-110)\*d/cos(atan(a2));

%line4平移后直线line\_4

b24 = b4 + (371-82)\*d/cos(atan(a2));

%求A2,B2,C2,D2坐标（line\_2,line\_4,line2两条垂线的交点）

%line\_4和垂线交点

syms xa2 xb2 b21\_1 b21\_2 x

b21\_1 = b21(1);

b21\_2 = b21(2);

x = solve('a2\*xa2+b24 = a4\*xa2+b21\_1', 'xa2');

xa2 = eval(x);

ya2 = a2\*xa2 + b24; % A2(xa2, ya2)

x = solve('a2\*xb2+b24 = a4\*xb2+b21\_2', 'xb2');

xb2 = eval(x);

yb2 = a2\*xb2 + b24; % B2(xb2, yb2)

%line\_2和垂线交点

syms xc2 xd2 x

x = solve('a2\*xc2+b2 = a4\*xc2+b21\_2', 'xc2');

xc2 = eval(x);

yc2 = a2\*xc2 + b2; % C2(xc2, yc2)

x = solve('a2\*xd2+b2 = a3\*xd2+b21\_1', 'xd2');

xd2 = eval(x);

yd2 = a2\*xd2 + b2; % D2(xd2, yd2)

%求AA2的垂直平分线

ka = (xa2-xa)/(ya-ya2);

ba = (xa^2-xa2^2+ya^2-ya2^2)/(2\*(ya-ya2));

%求BB2的垂直平分线

kb = (xb2-xb)/(yb-yb2);

bb = (xb^2-xb2^2+yb^2-yb2^2)/(2\*(yb-yb2));

%求旋转中心坐标

A = [ka; kb];

B = [-ba; -bb];

x0 = A\B;

y0 = ka\*x0+ba;

1. density.m

%J代表的是256\*256的还原图像的矩阵

a=zeros(0,10)

a(1)=J(256-26,46)

a(2)=J(256-88,64)

a(3)=J(256-111,84)

a(4)=J(256-115,193)

a(5)=J(256-124,142)

a(6)=J(256-128,193)

a(7)=J(256-143,196)

a(8)=J(256-168,95)

a(9)=J(256-204,46)

a(10)=J(256-252,111)

1. distance.m

%用于计算两种中心的偏移量

A=[]

for i=1:1:57

A(i)=floor(38.93\*cosd(33+i)+1.0)s

end

for i=148:1:180

A(i)=floor(38.93\*cosd(33+i)+1.0)

end

for i=58:1:147

A(i)=floor(-38.93\*sind(33+i)+1.0)

end

1. iradon.m

%A为原xls名，5为表格标签序号

S=xlsread('A',5)

J = iradon(S,34.5165:213.5165,'linear','Ram-Lak',1,512);

iptsetpref('ImshowAxesVisible','on')

figure

subplot(1,2,1)

imshow(J,[])

title('Filte1ed Backprojection')

1. standardize.m

iptsetpref('ImshowAxesVisible','on')

%18对应的是读取的xls文件，注意，这是裁剪行列后的xls文件，4代表的是标签序号

I1=xlsread('18',4)

J=imresize(I1,[256,256]);

for i=1:1:256

for j=1:1:256

J(i,j)=J(i,j)\*2.0023+0.0191;

%借助于分析中的参数，将其标准化

end

end

figure

subplot(1,2,1)

imshow(J,[])

title('Filte1ed Backprojection')

1. translate.m

%将毫米为单位的距离划为矩阵偏移量

sheet1=xlsread('A',4)s

position=[]

for i=1:1:10

for j=1:1:2

position(i,j)=round(sheet1(i,j)/100\*256);

end

end