实验四. 投影重建模拟实验

计算成像与统计学习 今天

实验目的

了解计算机断层成像中的重建算法的基本原理和深刻意义，理解正弦曲线的含义，掌握反投影重建图像方法。理解不同射线束、不同旋转角度等参数与图像重建质量之间的联系，及各自的优缺点。

实验内容

生成一幅phantom图像，使用radon变换，考察图像的正弦曲线图。

使用Radon反变换进行投影数据重建图像，考察不同参数下的重建效果。

给出一幅图象，试对其进行反投影重建。

比较采用笔束和扇束进行反投影重建的效果。

知识要点及参考程序

正弦曲线图

```
% 生成一个phantom图像，考察正弦曲线的数量及分布情况。
P = phantom(512);
figure, imshow(P)

theta = 0:180;
[R_512, xp] = radon(P, theta);
figure, imagesc(theta, xp, R_512)
xlabel('Parallel Rotation Angle - \theta (degrees)');
ylabel('Parallel Sensor Position - x\prime (pixels)');

% 生成一幅简单图像，考察正弦曲线的数量及分布情况。
f=zeros(256);
i = [23;103;254]; j = [11;124;252];
f(sub2ind(size(f), i, j))=1;
figure, imshow(f);

theta = 0:180;
[R_256, xp] = radon(f, theta);
figure, imagesc(theta, xp, R_256)
xlabel('Parallel Rotation Angle - \theta (degrees)');
ylabel('Parallel Sensor Position - x\prime (pixels)');
```

```
% 考察R_512和R_256矩阵的大小，分析尺寸含义及变化关系， R_512和R_256中存放的都是什么?

% 考察theta = 0:179；或者theta = 0:2:179;
```

Radon反变换投影重建

```
theta1 = 0:10:170;
[R1,xp] = radon(P,theta1);
num_angles_R1 = size(R1,2)          % 不同的旋转步进角度

theta2 = 0:5:175;
[R2,xp] = radon(P,theta2);
num_angles_R2 = size(R2,2)          % 不同的旋转步进角度

theta3 = 0:2:178;
[R3,xp] = radon(P,theta3);
num_angles_R3 = size(R3,2)          % 不同的旋转步进角度

% 考察一下    figure, plot(R3(:,1)), hold on        %    画出来的是什么?
          plot(R3(:,50))                  %    看到R3中存放的是什么了吗?



N_R1 = size(R1,1)
N_R2 = size(R2,1)
N_R3 = size(R3,1)

P_128 = phantom(128);
[R_128,xp_128] = radon(P_128,theta1);
N_128 = size(R_128,1)

figure, imagesc(theta1,xp,R1)
colormap(hot)
colorbar
xlabel('Parallel Rotation Angle - \theta (degrees)');
ylabel('Parallel Sensor Position - x\prime (pixels)');

figure, imagesc(theta3,xp,R3)
```

```
colormap(hot)
colorbar
xlabel('Parallel Rotation Angle - \theta (degrees)');
ylabel('Parallel Sensor Position - x\prime (pixels)');

%    分析一下theta1与theta2对应的正弦曲线之间的区别，是怎么造成的?
```

平行笔束与扇束反投影重建

```
% Constrain the output size of each reconstruction to be the same as the
% size of the original image, |P|.
output_size = max(size(P));

dtheta1 = theta1(2) - theta1(1);
I1 = iradon(R1,dtheta1,output_size);
figure, imshow(I1)

dtheta2 = theta2(2) - theta2(1);
I2 = iradon(R2,dtheta2,output_size);
figure, imshow(I2)

dtheta3 = theta3(2) - theta3(1);
I3 = iradon(R3,dtheta3,output_size);
figure, imshow(I3)
            % 考察不同旋转角度步进量对图像重建效果的影响，观察伪迹的形态

D = 250;
dsensor1 = 2;
F1 = fanbeam(P,D,'FanSensorSpacing',dsensor1);

dsensor2 = 1;
F2 = fanbeam(P,D,'FanSensorSpacing',dsensor2);

dsensor3 = 0.25;
[F3, sensor_pos3, fan_rot_angles3] = fanbeam(P,D,...
                                    'FanSensorSpacing',dsensor3);
```

```
figure, imagesc(fan_rot_angles3, sensor_pos3, F3)
colormap(hot)
colorbar
xlabel('Fan Rotation Angle (degrees)')
ylabel('Fan Sensor Position (degrees)')
                        % 考察扇形束投影的正弦曲线图
Ifan1 = ifanbeam(F1,D,'FanSensorSpacing',dsensor1,'OutputSize',output_size);
figure, imshow(Ifan1)

Ifan2 = ifanbeam(F2,D,'FanSensorSpacing',dsensor2,'OutputSize',output_size);
figure, imshow(Ifan2)

Ifan3 = ifanbeam(F3,D,'FanSensorSpacing',dsensor3,'OutputSize',output_size);
figure, imshow(Ifan3)
```

滤波反投影重建模拟

```
P = phantom(128);
R = radon(P,0:179);
I1 = iradon(R,0:179,'linear','none');
I2 = iradon(R,0:179);
figure,
subplot(1,3,1), imshow(P), title('Original')
subplot(1,3,2), imshow(I1,[]), title('Unfiltered backprojection')
subplot(1,3,3), imshow(I2,[]), title('Filtered backprojection')

% 自己调试一下采用不同的滤波器的重建效果差异
subplot(1,3,1), imshow(P), title('Original')
subplot(1,3,2),  imshow(iradon(R,0:179,'Ram-Lak'),[]),  title('R-L  filtered
backprojection')
subplot(1,3,3),  imshow(iradon(R,0:179,'Shepp-Logan'),[]),  title('S-L  filtered
backprojection')

% 自己找个简单图像重建试试
C = imread('cameraman.tif');
RC = radon(C,0:179);
figure,subplot(1,3,1), imshow(C,[]), title('Original')
subplot(1,3,2),  imshow(iradon(RC,0:179,'Ram-Lak'),[]),  title('R-L  filtered
```

```
backprojection')
subplot(1,3,3), imshow(iradon(RC,0:179,'Shepp-Logan'),[]), title('S-L filtered backprojection')
```