

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 课程报告



## CT 成像的扫描与重建算法实现

生物医学图像处理（2）课程作业

518021910971 裴奕博

## 目录

<b>1</b>	<b>项目背景</b>	<b>2</b>
1.1	CT 重建算法 . . . . .	2
1.2	数学基础 . . . . .	2
<b>2</b>	<b>项目实施</b>	<b>2</b>
2.1	项目总体实现 . . . . .	2
2.2	Radon 正变换的实现 . . . . .	3
2.3	Radon 逆变换的实现 . . . . .	4
2.4	图形化界面的实现 . . . . .	4
<b>3</b>	<b>项目结果和评价</b>	<b>5</b>
<b>4</b>	<b>感想与展望</b>	<b>5</b>

## 1 项目背景

### 1.1 CT 重建算法

自 20 世纪 70 年代被发明以来，X 射线计算机断层成像（CT）在医学影像检查中扮演了越来越重要的地位。在 CT 成像的流程中，将图像进行数字化、投影和重建的算法非常重要。CT 成像算法的好坏，会直接影响到 CT 的成像质量。

### 1.2 数学基础

中心切片定理和滤波反投影算法的提出，是 CT 成像和重建算法的数学基础。

中心切片定理可以表述如下：

**定理 1** (中心切片定理).

$$P(\omega, \theta) = F(\omega \cos \theta, \omega \sin \theta)$$

其中  $P$  是投影函数  $p(l, \theta)$  的傅里叶变换， $F(u, v)$  是图像的二维傅里叶变换。

该定理说明，某断层在角度为  $\theta$  时得到的平行投影的一维傅里叶变换，等于图像二维傅里叶变换过原点的一个垂直切片，且切片的方向也为  $\theta$  角。

通过中心切片定理，我们可以得到原 CT 图像的投影图，再经过重建算法就可以得到 CT 重建图像。重建算法主要分为两大类：滤波反投影算法和反卷积反投影算法。本次的重建算法属于滤波反投影算法，可以用公式表示如下：

**定理 2** (滤波反投影算法).

$$f(x, y) = \int_0^\pi |\omega| P(\omega, \theta) \exp(j2\pi\omega l) |_{l=x \cos \theta + y \sin \theta}$$

其中  $P(\omega, \theta)$  是通过中心切片定理得到的投影图的傅里叶变换。

## 2 项目实现

### 2.1 项目总体实现

项目的整体流程比较简单，如下图：

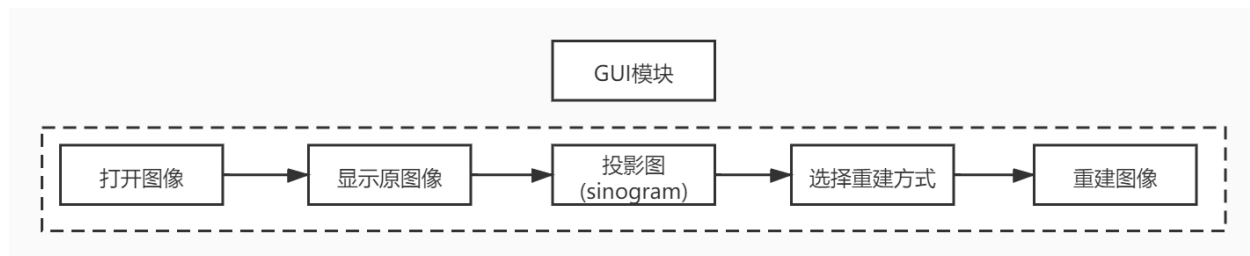


图 1: 项目流程图

所有程序均采用 Python 实现，图形化界面使用 PySide2 图形化框架实现。图像处理与重建的部分使用 numpy 实现，所使用的的依赖包可见 requirements.txt 文件。

## 2.2 Radon 正变换的实现

Radon 正变换的实现方法如下图：

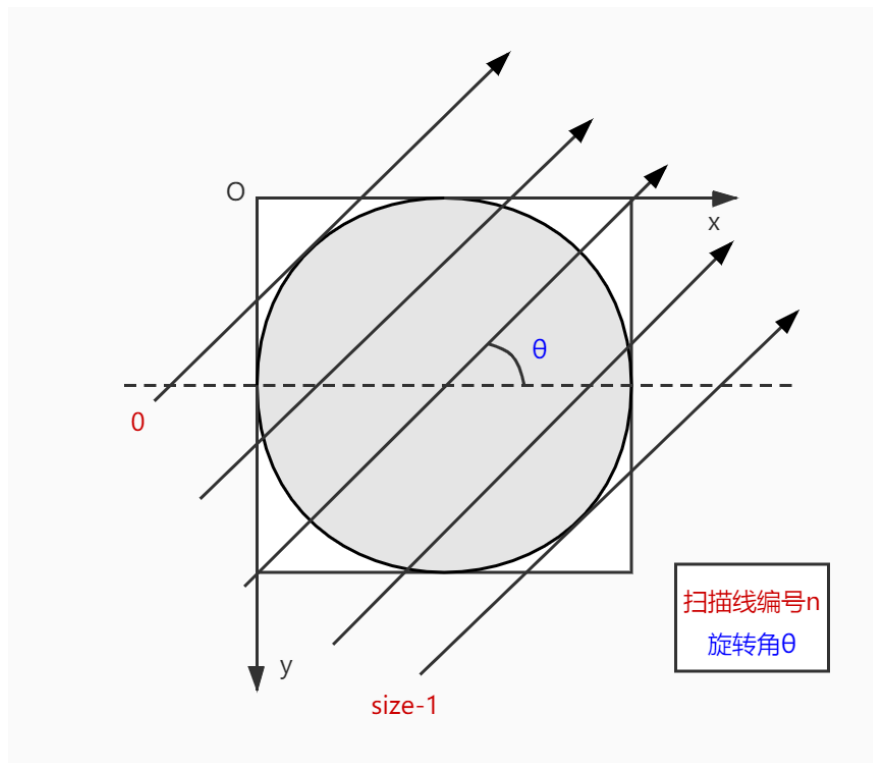


图 2: Radon 正变换实现

其中最大的正方形表示待扫描的图像。我们默认扫描的区域范围用灰色部分显示。默认的扫描线方向如图中虚线所示，编号从上到下为  $[0, \text{size}-1]$ 。扫描线与默认方向之间的夹角为  $\theta$ 。设起始扫描线的坐标为  $(x, y)$ ，则

$$1324 \quad (1)$$

最后将每条扫描线上的值相加，就可以得到其中一个  $\theta$  角所对应的扫描值，将其显示出来即可得到正弦图。

## 2.3 Radon 逆变换的实现

## 2.4 图形化界面的实现

图形化界面采用了 PySide2 (PyQt) 框架实现。UI 文件储存在 ui 文件夹下，由两个页面组成，一个主页和一个滤波反投影算法的参数选择界面。

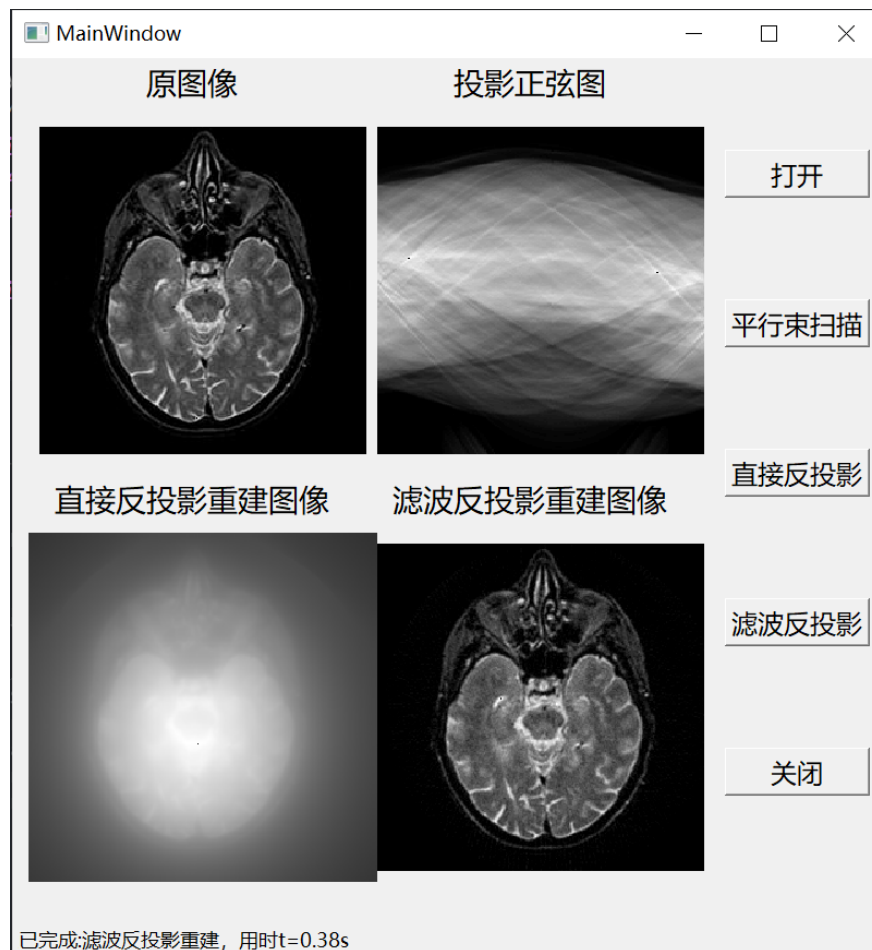


图 3: 主界面

本次实现的图形化界面具有以下几个特点：

- 可以自行选择本地文件打开。
- 使用了多线程操作。图像处理等耗时的操作运行在另一个线程下，但是在计算过程中将页面上所有交互按钮禁用，防止用户进行误操作。
- 底部有状态栏，在运行过程中可以给予提示，运行结束后可以显示进行的操作和所消耗的时间。
- 在用户误操作时可以跳出错误提示。

### 3 项目结果和评价

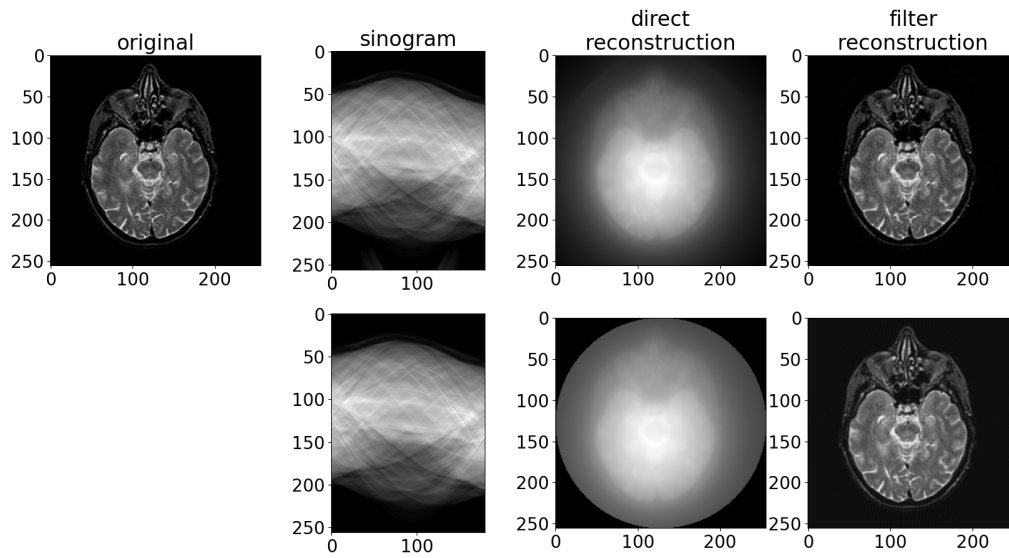


图 4: CT 图像重建结果

### 4 感想与展望