**简易DR图像浏览器设计技术报告**

# 一、软件介绍

简易DR图像浏览器是针对TIF格式的医学图像浏览而设计的一款软件，提供可视化界面进行操作，上手简单。主要功能有：图像文件的读取、查看图像信息、窗宽窗位调整、图像缩放、图像反相、图像翻转、图像增强、修改后的图像保存以及一系列组合操作。

DR简易图像浏览器的主界面样式如下图所示：



**图 1 简易DR浏览器主界面**

# 二、软件需求分析

**1、读取给定格式的医学图像文件**

给定实验图像像素灰度值12位有效，与常规的8位灰度图像不同，对图像进行矩阵存储时需要更大的空间。

**2、灰度窗宽窗位调整**

由于普通显示器无法直接将4096级有效的像素值进行显示，因此需要进行窗宽窗位调整，根据窗位与窗框对4096级的像素值进行映射，从而实现在不同窗宽窗位下对图像全部像素值进行显示。

为了方便进行窗宽窗位调整，编写的程序利用的是“滑条”来实时地进行调整窗位窗宽，这样能够更为方便快捷地进行调整，也能够实时地看见调节的效果。

|  |  |
| --- | --- |
|  |  |
| **图 2 利用滑块实现的窗宽窗位调整按钮** | |

**3、灰度反转与图像左右翻转**

灰度反转和图像翻转功能均是对直接进行图像观察功能的补充，实现对图像信息更完整的获取。

**4、图像缩放**

在观察医学图像时，有时需要对可能存在病灶的区域进行局部放大，进行细致观察，因此需要实现图像缩放功能，同时需要通过特定算法实现插值，来保证缩放后所观察的图像依旧有较高的观察质量。

为了避免出现插值后显示的图像过大溢出显示器范围的情况，本次实验中未采取根据滚轮等操作实时改变大小，而是输入固定倍率，待凸显显示后可以通过面板上的“x/y坐标调整”对图像的不同部位进行查看。此外，插值算法运算过程运算较慢，采用上述的方式也能提升查看响应速度，提升用户体验。

|  |  |
| --- | --- |
|  |  |
| **图 3 利用滑块实现的不同缩放区域的浏览** | |

**5、细节增强**

对于可能存在模糊的部分，需要通过细节增强算法实现图像锐化等操作，从而获得更为细致清晰的观察效果。

# 三、系统设计

**1、界面设置**

整体分为“上、中、下”三块，上半部分为对于图像的“读取、保存、信息获取以及重置”，为图对图像的基本操作区域；中间部分是程序的主要功能区，从左至右依次包含“窗宽窗位调整、图像缩放、灰度反转与图像翻转、细节增强”；下半部分主要为对于程序的使用做了一点简单说明与提醒。

**2、部分功能与算法设计**

采用前后端分离的设计，并且将处理过程均包装进函数中进行调用，尽可能提高代码的可阅读性以及可维护性。

**图像信息维护。**在读取到文件信息self.img之后，我们将其进行深拷贝得到其备份数据self.temp，后续的数据操作都是在对self.temp的基础上进行实时修改。

**重置功能。**在每次执行图像处理前（如窗宽窗位调整）都会对当前的图像进行备份，以便于在执行图像操作后复原至操作前的图像。此外还包含一个全局重置功能，恢复到新读取图像时的状态。

**操作可叠加。**如先进行窗宽窗位调整，再进行灰度反相，再对反相后的图像进行方法查看细节，最后对感兴趣的区域进行保存。

**窗宽窗位调整。**根据从UI界面获得的值进行实时调整与显示，通常情况下根据图像的像素值设置默认的窗位窗宽进行显示。

**图像缩放。**采用双线性插值算法进行图像缩放操作，首先进行图像对齐，以保证源图像和目标图像几何中心对齐，并且对图像边缘的像素点做了特殊处理，防止出现边界实现无法插值的情况。

**细节增强。**利用均值滤波算法获取平滑后的图像，从而实现Unsharp Msaking算法。

# 四、软件功能测试与对比验证

**1、实验文件读取与信息显示（foot.tif）**

|  |  |
| --- | --- |
|  |  |
| **图 4 当前图像在默认窗位窗宽下显示** | |

**2、窗宽窗位调整测试（lung.tif）**

|  |  |
| --- | --- |
| 简易DR浏览器显示图像 | ImageJ显示图像 |
|  |  |
| Window\_level = 2061, window\_wide = 4068 | Window\_level = 2061, window\_wide = 4068 |
|  |  |
| Window\_level = 2038, window\_wide = 4068 | Window\_level = 2038, window\_wide = 4068 |
|  |  |
| Window\_level = 2038, window\_wide = 3798 | Window\_level = 2038, window\_wide = 3798 |
| **图 5 不同窗宽窗位下的显示效果对比** | |

根据实验结果显示，简易DR浏览器实现的图像显示功能与专业软件ImageJ所显示的图像基本一致，且功能正常。

**3、图像缩放测试（lumbar.tif）**

|  |  |
| --- | --- |
| 简易DR浏览器显示图像 | 将原始图像放大2倍（原图左上角） |
|  |  |
| 调整x坐标（约40） | 继续调整y坐标（约80） |
|  |  |
| **图 6 图像放大后不同区域分别滑动显示** | |

实验结果显示可以通过对滑块的移动对放大后不同区域的图像进行浏览，功能正常，除了计算插值过程中耗时较长外，拖动浏览时响应速度较快。

**4、灰度反转与图像翻转（pelvis.tif）**

|  |  |
| --- | --- |
| 简易DR浏览器显示图像 | 简易DR浏览器灰度反转后的图像 |
|  |  |
| 简易DR浏览器左右翻转后的图像 | ImageJ左右翻转后的图像 |
|  |  |
| **图 7 灰度反转与图像左右翻转** | |

从上图中可以看到可以实现图片反相和左右翻转，且左右翻转结果与ImageJ结果一致。

**5、图像增强（pelvis.tif）**

|  |  |
| --- | --- |
| 简易DR浏览器显示图像 | 均值核大小：3，残差权重：1.0 |
|  |  |
| 均值核大小：3，残差权重：2.0 | 均值核大小：5，残差权重：1.0 |
|  |  |
| **图 8 不同参数设置下的图像增强结果显示** | |

放大图片可以看到进行细节增强后的图像可以实现将图像边缘细节进行锐化的效果，使得图像显得更加清晰。

# 五、实验小结

本次实验所制作的程序能够完全实现实验所需功能，但整体功能还略显匮乏，如在进行窗宽窗位调整时可以增加灰度直方图的显示，增加边缘提取功能等。

本次实验历时两周，卡壳最多的部分为算法到实现的转换，对于代码经验非常匮乏的我而言，经常出现未知原因的小bug，如深浅拷贝带来的值的变化不同，python中列表作为函数形参使用后列表值会改变等等；但也收获了很多，如将双线性插值算法进行了实现，更深入理解了算法的具体含义与操作步骤；对部分操作功能进行了优化，加强用户体验；以及从零开始学会绘制简单的UI界面并与后端算法进行连接，这大大锻炼了我的动手能力，也巩固了在课上所学习知识。