图像细节增强实验报告

09019204 曹邹颖

一. 实验内容

本实验要求扩展实验 2 (灰度映射实验),设计图像增强处理流程,实现图像细节增强,同时要求增强后的图像噪声没有明显放大。增强后的图像可以调用实验 2 的功能显示。

二. 实验要求

- 1. 程序使用 C++语言编写,集成开发环境可以选择 vs 或 Qt, 灰度变换的核心代码不允许直接调用库函数实现,不允许使用 opencv 等第三方库。
- 2. 实验图像数据有效灰度范围[0,4095],即 12 位有效灰度,每像素 2 字节(最高 4 位数据无效,有效灰度保存于低 12 位)。数据文件为自定义格式(非标准格式),文件中的数据存放如下图:

文件开始的 4 字节存放图像宽,其后 4 字节存放图像高,此两参数均为无符号长整型(unsigned long),紧随其后为按光栅扫描顺序(从左向右,逐行扫描)存放的像素值,像素值为无符号短整型(unsigned short)。所有多字节数据都按 intel 顺序(即低字节在前,高字节在后)存放。文件不包含其它数据。

图像增强的实验结果应包含三幅截屏图像(以下灰度窗的位置记为[窗位, 窗宽], 其中窗位表示灰度窗中央位置的灰度值, 窗宽表示灰度窗的宽度): [1] lung.raw 文件包含的图像, 以灰度窗[2048, 4096]显示, 并将胸脊(胸椎)置于画面的中间位置; [2] lung.raw 文件包含的图像, 以灰度窗[3000, 2000]显示, 并将肺(左、右肺任选)置于画面的中间位置; [3] knee.raw 文件包含的图像, 以灰度窗[250, 500]显示, 并将膝关节置于画面的中间位置。

3. 自由设计图像增强流程和处理流程中的算法。

三. 实验设计

1. 处理流程的详细描述(包括流程框图)

为保证图像细节增强的同时,图像噪声不会被明显放大,想到的思路是先进行一次图像平滑处理,再进行图像锐化处理。

设计了一个 CImageEnhancement 类继承自 CImageData 类,用于存放图像增强后的图像数据,调用图像平滑算法、锐化算法。其中图像平滑算法包括 void Gaussian(int wSize, double sigma, CImageData<T_DATA>*);高斯滤波器函数(自定义滤波器窗口大小和参数σ),void AveragingFilter(int wSize, CImageData<T_DATA>*);均值滤波器函数(自定义滤波器窗口大小),void MedianFilter(int wSize, CImageData<T_DATA>*);中值滤波器函数(自定义滤波器窗口大小),锐化算法选择的是 void Laplace(double k, CImageData<T_DATA>*);利用 Laplace 算子的图像锐化函数(自定义二阶微分前的权重参数 k)。

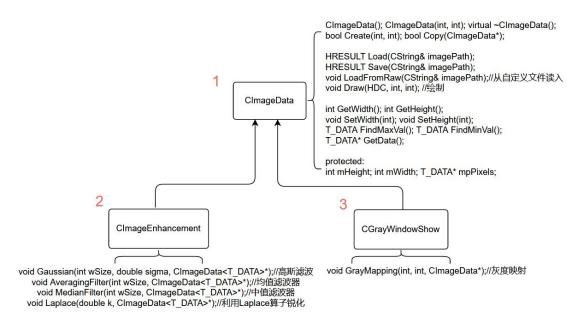


图 1. 实验相关函数框图

实验流程: 首先在菜单中选择"灰度窗浏览图像",弹出灰度窗对话框,点击"载入灰度图像"按钮,弹出文件对话框 CFileDialog,选择图像文件所在路径,选择确认从而将图像数据读入 CImageData 类对象 rawImage 中。接着,点击"图像增强"按钮,弹出图像增强对话框,首先选择图像平滑方式(也可以不选直接进行图像锐化),设置高斯滤波器单选按钮 Radio_Gaussian 的 Group 属性,关联 int 型变量 Filter_Radio,使得三个 Radio Button 组件成组使用,相互排斥只能选其一,三个按钮分别绑定一个事件处理程序,用来调节 int 型变量 Filter_Radio 的值,=0 代表选择高斯滤波器,=1 代表选择均值滤波器,=2 代表选择中值滤波器,选完图像平滑方式后输入需要的自定义滤波器窗口大小或者权重参数,之后再设置图像锐化算法中 Laplace 算子前的权重参数 k,点击确认便会将 rawImage 中的原始数据处理后存到 CImageEnhancement 类对象 enhancedImage 中。最后,输入窗宽窗位,会将 enhancedImage 中的图像数据灰度映射后显示出来。

如果,图像增强效果不明显,可以再次点击图像增强按钮,重新选择图像平滑方式亦或重新设置窗口大小或者图像锐化算法的权重参数,再通过灰度窗浏览比较前后变化,注意这里就不需要再次载入灰度图像了,因为原始图像保存在 rawImage 中并没有发生改变,图像增强的处理只是改变了 enhancedImage 中的图像数据。



图 2. 实验人机交互窗口流程框图

2. 处理流程中采用的算法

① void Gaussian(int wSize, double sigma, CImageData<T_DATA>*); 高斯滤波器函数(自定义滤波器窗口大小和参数σ)

高斯滤波器的传递函数是高斯函数,当 wSize=2k+1 时, $H_{ij} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}}$;

② void AveragingFilter(int wSize, CImageData<T_DATA>*); 均值滤波器函数(自定义滤波器窗口大小)

均值滤波器是在它的空间域里对它命名的,点扩展函数为 $\frac{1}{\text{wSize*wSize}}$ · $\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$

窗口越小,对噪声抑制能力越差,但是计算复杂度小;

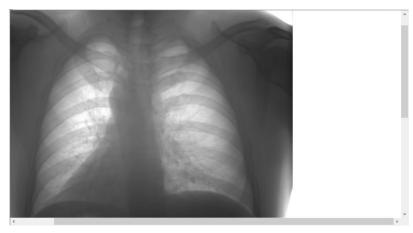
- ③ void MedianFilter(int wSize, CImageData<T_DATA>*); 中值滤波器函数(自定义滤波器窗口大小) 通过将 wSize*wSize 窗口内的像素值排序,选出中值作为窗口中心的像素值,非线性方法,计算复杂度稍高,且窗口宜小,否则小于 ½ 窗口的细节将会被滤掉;
- ④ void Laplace(double k, CImageData<T_DATA>*); 利用 Laplace 算子的图像锐化函数(自定义二阶微分前的权重参数 k) 由于是数字图像,可以算出一阶微分 $\nabla f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} = 2f(x,y) f(x-1,y) f(x,y-1),二 阶微分<math>\nabla^2 f = 4f(x,y) f(x-1,y) f(x,y-1) f(x+1,y) f(x,y+1)$,从而利用 $\hat{f}(x,y) = f(x,y) k \cdot \nabla^2 f(x,y)(k > 0)$ 公式进行图像锐化。

四. 实验结果

针对实验给出的两幅灰度图像,

1. lung.raw 文件包含的图像,以灰度窗[2048, 4096]显示,并将胸脊(胸椎)置于画面的中间位置;

原图像:



选择 wSize=3, σ =0.8 的高斯滤波+k=6 的 Laplace 算子权重参数进行锐化得到:



图像增强后

选择 wSize=3 的均值滤波+k=6 的 Laplace 算子权重参数进行锐化得到:

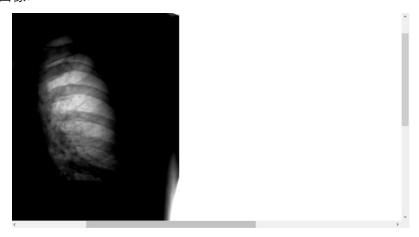


选择 wSize=3 的中值滤波+k=6 的 Laplace 算子权重参数进行锐化得到:

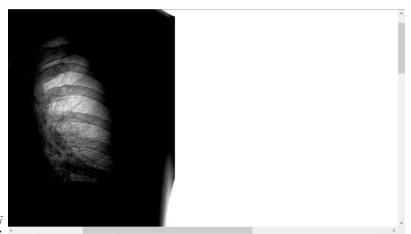


2. lung.raw 文件包含的图像,以灰度窗[3000, 2000]显示,并将肺(左、右肺任选)置于画面的中间位置;

原图像:

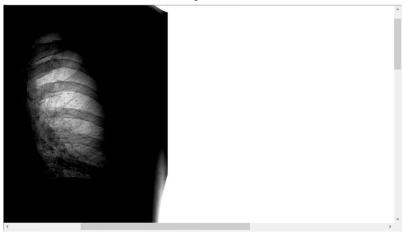


选择 wSize=3, σ =0.8 的高斯滤波+k=6 的 Laplace 算子权重参数进行锐化得到:

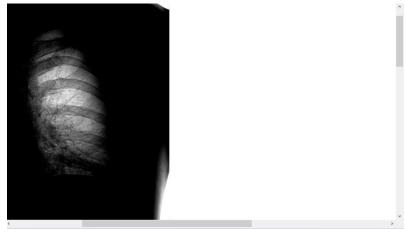


图像增强后

选择 wSize=3 的均值滤波+k=6 的 Laplace 算子权重参数进行锐化得到:



选择 wSize=3 的中值滤波+k=6 的 Laplace 算子权重参数进行锐化得到:



3. knee.raw 文件包含的图像, 以灰度窗[250, 500]显示, 并将膝关节置于画面的中间位置。



选择 wSize=3, σ =0.8 的高斯滤波+k=2 的 Laplace 算子权重参数进行锐化得到:



选择 wSize=3 的均值滤波+k=2 的 Laplace 算子权重参数进行锐化得到:

图像增强后



选择 wSize=3 的中值滤波+k=2 的 Laplace 算子权重参数进行锐化得到:



由实验图像结果可见, 在滤波器窗口较小的情况下, 不同的图像平滑方法再进行一致的图像锐化算法, 实验结果细节增强并没有太多差距。不过, 从 UI 的反应时间可看出中值滤波器的计算复杂度比高斯滤波器、均值滤波器的计算复杂度要大。

但总的来说, 先做图像平滑再做图像锐化的实验流程是可以肯定的, 能较清楚地人眼看出处理后图像的细节相比原图像得到增强, 且没有引入明显的噪声。