浙江大学实验报告

姓名：林炬乙　　学号：3180103721

课程名称：数字图像处理　　任课老师：项志宇

实验名称：最近邻、双线性和双三次方插值　　实验日期：2021/3/20

1 实验目的和要求

（分点简要说明本次实验需要进行的工作和最终的目的）

将一幅 Lina 图像降采样后，分别利用最近邻、双线性和双三次方插值恢复至原分辨率，并比较效果差异。要求除了图像输入和输出显示外，核心处理函数自己编写，不能直接调用 openCV 或者 matlab 里的相关函数。

2 实验原理

• 内插是图像放大、收缩、旋转和几何校正等任务中广泛使用的 基本工具。是一种基本的图像重取样方法。 • 是用已知数据来估计未知位置的数值的处理

数字图像的质量在很大程度上取决于取样和量化中所用的样本数和灰度级 。

上采样就是放大图像，通过放大图像，我们可以增加图像的像素。降采样就是缩小图像，通过缩小图像，我们可以减少图像的像素，一方面，我们可以生成对应图像的缩略图，另一方面，我们可以减少图像的大小，节约内存。

降采样: 已有库为 pyrDown ,pyrUp,可以作为参考.

pyrUp 函数，首先对输入图像进行上采样，在图像的偶数行和偶数列插入 0；然后进行滤波。pyrDown 函数正好相反，先对输入图像进行滤波，然后剔除图像的偶数行和偶数列。

• 最近邻内插 • 把原图像中最近邻的灰度赋给每个内插的新位置.

• 双线性内插(用 4 个最近邻) • V(x,y)=ax+by+cxy+d 注意：实质上是一种非线性内插方法

双三次内插（用 16 个最近邻，保持细节更好）

基于 BiCubic 基函数的双三次插值法

双三次插值又称立方卷积插值。三次卷积插值是一种更加复杂的插值方式。该算法利用待采样点周围 16 个点的灰度值作三次插值，不仅考虑到 4 个直接相邻点的灰度影响，而且考虑到各邻点间灰度值变化率的影响。三次运算可以得到更接近高分辨率图像的放大效果，但也导致了运算量的急剧增加。

假设源图像 A 大小为 m*n，缩放后的目标图像 B 的大小为 M*N。那么根据比例我们可以得到 B(X,Y)在 A 上的对应坐标为 A(x,y)=A(X*(m/M),Y*(n/N))。在双线性插值法中，我们

选取 A(x,y)的最近四个点。而在双立方插值法中，我们选取的是最近的 16 个像素点作为计算目标图像 B(X,Y)处像素值的参数。P 点就是目标图像 B 在(X,Y)处对应于源图像中的位置，P 的坐标位置会出现小数部分，所以我们假设 P 的坐标为 P(x+u,y+v)，其中 x,y 分别表示整数部分，u,v 分别表示小数部分。那么我们就可以得到如图所示的最近 16 个像素的位置，在这里用 a(i,j)(i,j=0,1,2,3)来表示。双立方插值的目的就是通过找到一种关系，或者说系数，可以把这 16 个像素对于 P 处像素值影响因子找出来，从而根据这个影响因子来获得目标图像对应点的像素值，达到图像缩放的目的。这种算法需要选取插值基函数来拟合数据，其最常用的插值基函数也就是本次实验采用 BiCubic 函数作为基函数。

3 实验内容

（分点阐述实验步骤）

1. 安装 openCV

2021 年 3 月 20 日星期六 中午 12:43 完成

2. 写降采样函数

伪代码如下

```
    k = 0 ;
    for(int i  =0  ; i <  rows ; i = i + 2){
   行      newimg[k++] =  srcimg[i];
    }
    k = 0;
    for(int i  =0  ; i <  cols ; i = i + 2){
  列       resimg[k++] =  newimg[i]  ;
    }
```

3. 写最近邻内插

每一行,比例放大, 四舍五入取 int , 获得原来的最近点.

目标各像素点的灰度值代替源图像中与其最邻近像素的灰度值。

4. 写双线性内插

首先在 x 轴方向上，对 R1 和 R2 两个点进行插值，这个很简单，然后根据 R1 和 R2 对 P 点进行插值，这就是所谓的双线性插值。（f (*)为 * 点处像素值）

在 x，y 方向分别进行插值计算，最后对 P 点进行插值计算：

线性插值的结果与插值的顺序无关。首先进行 *y* 方向的插值，然后进行 *x* 方向的插值，所得到的结果是一样的。双线性插值的结果与先进行哪个方向的插值无关。

如果选择一个坐标系统使得 的四个已知点坐标分别为 (0, 0)、(0, 1)、(1, 0) 和 (1, 1)，那么插值公式就可以化简为

f(x,y)=f(0,0)(1-x)(1-y)+f(1,0)x(1-y)+f(0,1)(1-x)y+f(1,1)xy

5. 写双三次方插值

写计算系数的函数, 选择四个点, 然后根据公式计算系数.

写三次插值.

这个效果好像不是很好, 可能是多项式选择的问题.

一开始写的非常麻烦, 非常的长.系数计算.

后来写在嵌套 for 循环中，两个数组下标取值，就不用把 16 个坐标一个个乘起来了，这样又方便又不容易错.

遇到的错误:

问题 0. 0x00007FFE4143D759 处(位于 Project1Nearest neighbor, bilinear and bicubic interpolation.exe 中)有未经处理的异常： Microsoft C++ 异常：cv::Exception，位于内存位置 0x000000D22A9DF1B0 处。
原因：位置不对，Lina 照片要和 cpp 文件放在同一个文件夹,不是和.sln 文件放在同一个文件夹.

正确: "./Lina.jpg" 可以 ， "Lina.jpg" 也可以.

问题 1 `pyrDown ,pyrUp` 不存在

正确:包含#include <opencv2/imgproc/imgproc.hpp> 即可，但是我们要自己写

问题 2 `static_cast<int>(i / kx + 0.5)` 会出错

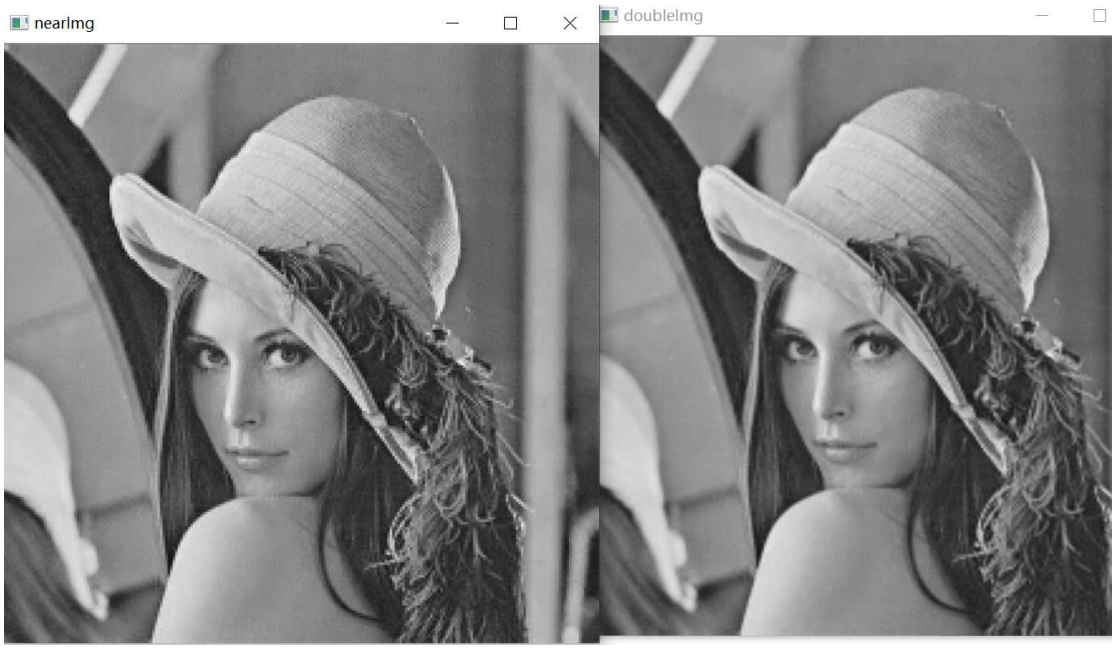
用 `x = static_cast<int>((i + 1) / kx + 0.5) - 1;` 可以
因为他可能超出边界，所以要-1;

问题 3 还是异常,他不显示更详细的错误，就显示异常．改成循环就出错．
因为我的循环写错了．

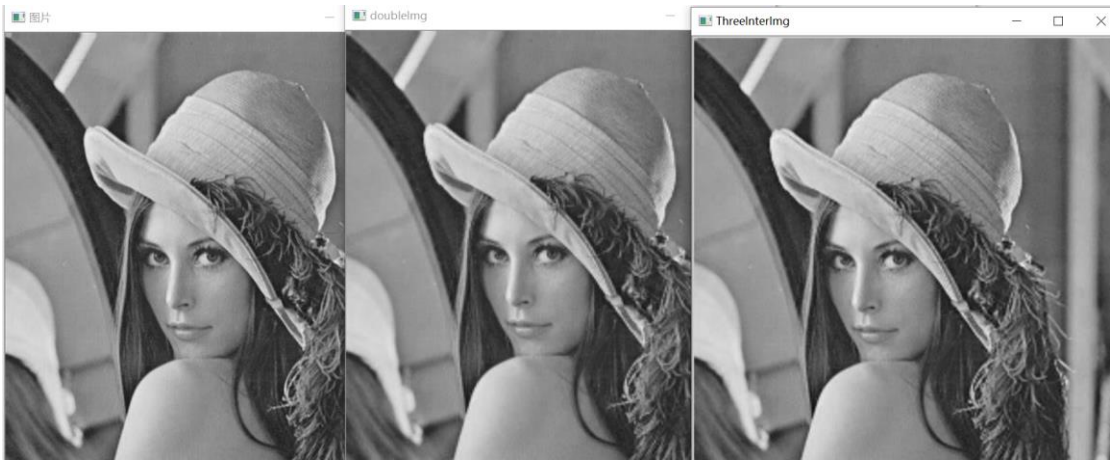
4 实验结果和分析

（使用图片和文字叙述实验结果，并对这些结果进行适当分析）

直接内插和原图相比可以看出，肩膀皮肤边缘明显有颗粒,鼻子有棋盘格图形.



可以看出，双线性内插后鼻子部位的马赛克,和脸部边缘的马赛克就没有最近邻内插的图像那么明显,说明恢复效果较好.



二次插值和三次插值似乎区别不大. 可能原图本身就分辨率比较低，还原后分辨率不高.