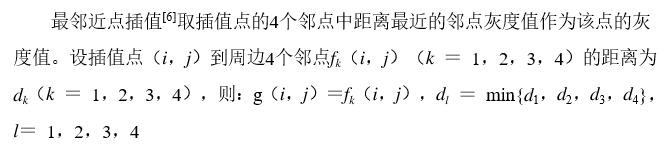
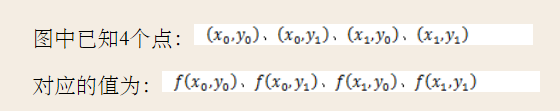
**1 最邻近点插值 nearest\_interpolation.cu**

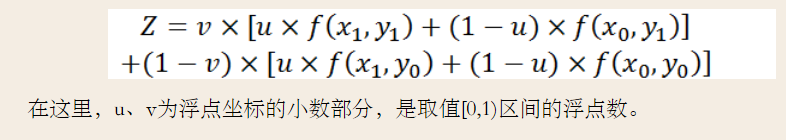
该法针对于二维图像取待采样点周围4个相邻像素点中距离最近的1个邻点的灰度值作为该点的灰度值。

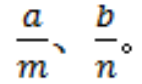


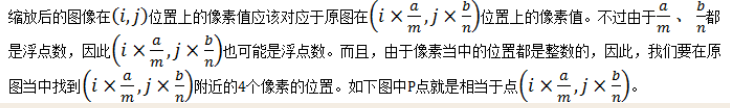
**2 双线性**：Bilinear\_Interpolation.cu

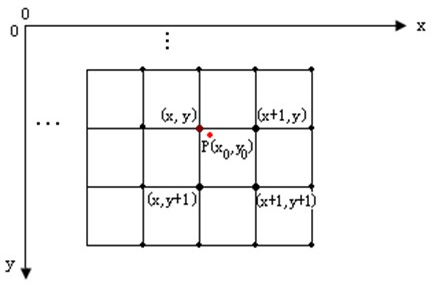
双线性插值是利用了需要处理的原始图像像素点周围的四个像素点的相关性，通过双线性算法计算得出的。对于一个目的坐标，通过向后映射法得到其在原始图像的对应的浮点坐标（i+u,j+v），其中I,j均为非负整数，u,v为[0,1]区间的浮点数，则这个像素的值f(i+u,j+v)可由原图像中坐标为(I,j),(i+1,j),(I,j+1),(i+1,j+1)对应的周围四个像素的值决定





设原图（a,b），缩放后图（m,n），那么原图与缩放后的图在xy方向上的缩放比为

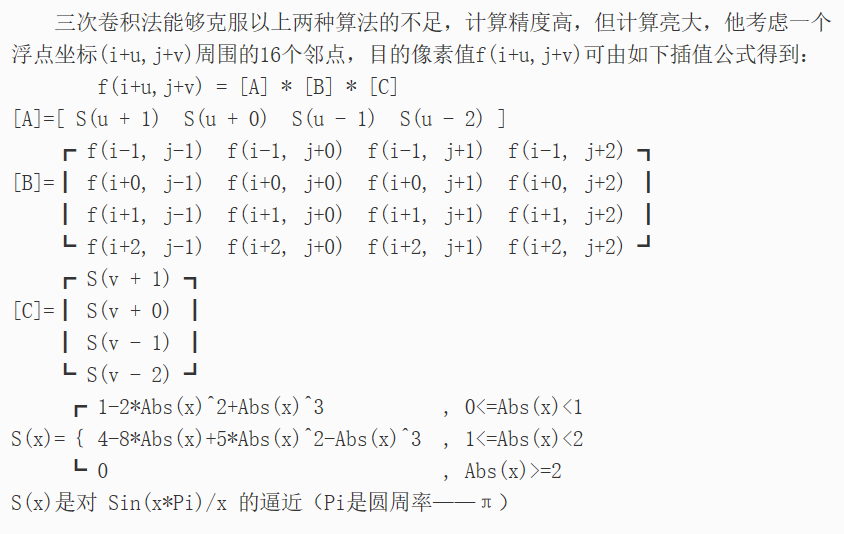


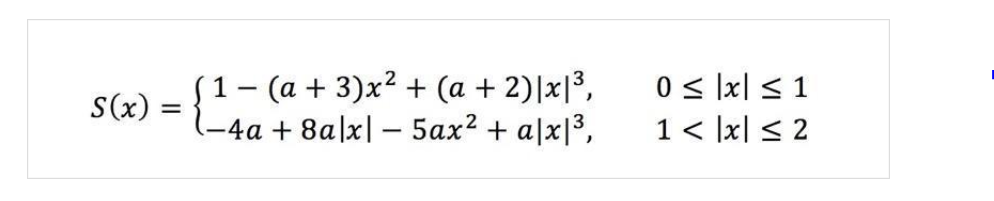


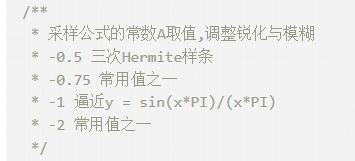
参考网站（有代码）：<https://www.cnblogs.com/zzw-in/p/Bilinear_interpolation.html>

http://blog.csdn.net/zyttae/article/details/42710303

**3. 双立方卷积法（二维三次卷积）Cubic\_Convolution\_Interpolation.cu**







**4.双立方卷积法加极值 Cubic\_Convolution\_Interpolation\_range.cu**

在双立方卷积法的基础上增加一个功能，如果变换后像素点对应原始像素点附近四个点的极值<10，那么这个点不进行双立方卷积法，使用这四个点的平均值来计算。

对于三通道的彩色图像，则是处理后图像中的点对应原始图像中的点，三个通道的极值都<10，那么才进行平均值计算。

**5.欧式距离插值**

a. 找到待插值点的原图像对应点M

b. 取原图像3\*3领域的点，计算每个点到M的加权欧式距离D=

，W是的累加

是i到M的欧式距离，（，先做一次实验，各做一次缩小0.5倍）

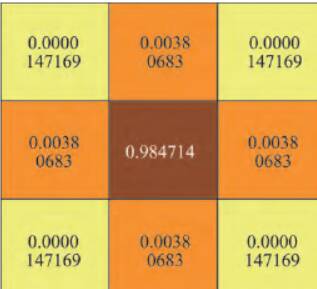
c. 待插值点的像素值为：DN=

D是所有点加权欧式距离的总和

是i点的像素值。

**6.邻域内计算 distance\_Interpolation\_6.cu**

在第五个实验的基础上稍微改一下，做一个实验。在找到3\*3的邻域后，不算距离了，用这个模板乘以对应位置的像素值，得到待插值的像素值。然后效果图和还原图。



**附加算法：将变换后的图像，返回并覆盖原始图像对应的像素点。 return.cu**