今天和大家掰扯一个特别简单的概念！我也不知道有没有用，但是对像我一样的小迷糊还是应该有帮助的。

前一段时间帮导师做了一个任务，其中要用到双线性插值。虽然之前学过，而且这个方法也很容易理解，但是我从来没有在实际中用过。所以，还是由于理解的不够透彻而让我在使用时造成了很大的困扰。

前天的《数字图像处理》课刚好讲到了插值，所以整理一下这部分的知识。

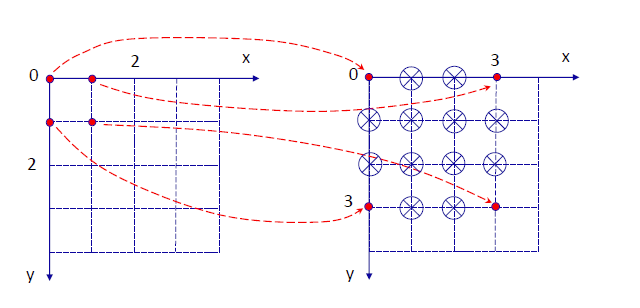
**错误的观念**

其实插值的概念非常简单。但是就是由于它太简单了，我一直没有认真理解过，也没有人纠正，所以我对其存在一些根深蒂固的误解。

先说说我之前的错误理解吧。我们都知道，将图片放大后，有一些像素，我们是找不到他对应的值的。例如下图中右面的图片中的 [0,1]，[0,2] 等等。我一直认为双线性插值的意思是：右图中 [0,1] 的值是由右图中 [0,0]，[0,3]，[3,0]，[3,3] 四个点的值通过计算得出的。

其实，这个想法是大错特错的。如果有的小伙伴和以前的我一样迷糊的，可能就会想，怎么会不对呢？

要讲清楚这个问题，就要从如何放大一张图片说起。



将图片放大3倍

**图片放大**

我们想得到一张放大的图片，现在有原图像（srcImage）和目标图像（dtsImage）。那么，有一个最基本的问题摆在我们面前：是遍历 srcImage 呢，还是遍历 dstImage 呢？（咋还整出代码来了呢？）

在实践的过程中，通常都是遍历 dstImage 的。因为这样可以确保 dstImage 的每一个像素都是有值的。

就拿上图的例子来说，右图（也就是 dstImage）中的 [0,0] 点很顺利的找到了左图中与自己对应的 [0,0] 点。然后 [0,1] 点就懵逼了：“我应该找 [0,0.33]，也没有这个点啊！”

插值这个方法，就是为了解决这个问题的。

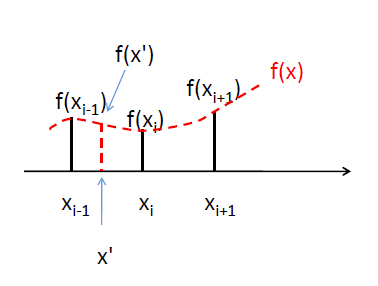
**插值**

**插值（Interpolation）**，按我们之前理解的就是在两个数中间插一个数呗。不过，老师说插值问题实际上是一种 ***拟合问题***。

我之所以写这篇文章有一半的原因是因为这句话，因为我觉得这为我们的理解又提供了一种新思路。那么，怎样理解这句话呢？

还是以上一节中的例子为例。下面这幅图描述的是以为的情况，就相当于把上一节 srcImage 的第0行拿出来。f(xi-1) 是 [0,0] 的灰度值，f(xi) 是 [0,1] 的灰度值，f(xi+1) 是 [0,2] 的灰度值。f(x') 就是我们想得到的 [0,0.3] 的灰度值。

所以，所谓插值就是用 x' 某个领域内的函数值按照一定规则拟合出一个函数，再在其中查找 f(x') 的值。



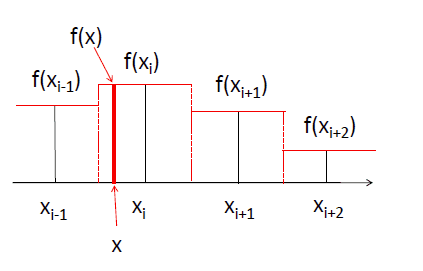
插值

**几种插值方法**

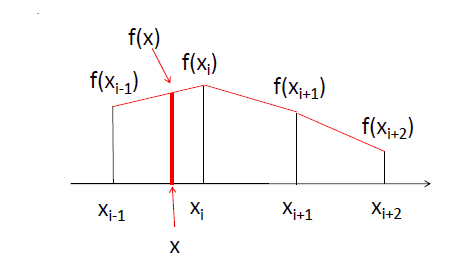
我觉得这几种方法都不用解释了，就直接放图就可以了。  
（好敷衍呐！）（闭嘴吧……）

不过这几张图还是很好的，很清晰明了。

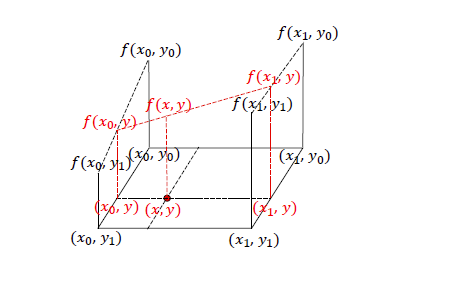
**1. 最近邻插值（Nearest Interpolation）**



**2. 线性插值（Linear Interpolation）**



**3. 双线性插值（Bilinear Interpolation）**



基于上一节的理解，我们就可以设定任意规则去拟合，然后做插值了！可以用邻域的三个点、四个点拟合，甚至可以用二次曲线、三次曲线……（只要不挨打就行）

**总结**

写到这儿，我最开始犯的错误就很明显了。因为插值是对 srcImage 中某个邻域内进行插值，从而得到 dstImage 中某个像素值。而不是在 dstImage 中插值。

有些小伙伴可能会说，那这两种操作的值不是一样的吗？

针对放大、缩小这种变换，确实是一样的。但是如果是图像旋转、甚至扭曲呢？这就完全无法用 dstImage 中的点去插值了。（别问我咋知道的……）

**最后**

既然是从实践中发现的问题，就在实践中结束。下面附上一段 OpenCV 双线性插值的核心代码。

//整数部分

int xi = (int)newx;

int yi = (int)newy;

//小数部分

double xd = newx - xi;

double yd = newy - yi;

xi = xi > 0 ? xi : 0;

xi = xi < colNumber - 1 ? xi : colNumber - 2;

yi = yi > 0 ? yi : 0;

yi = yi < rowNumber - 1 ? yi : rowNumber - 2;

Vec3b index00 = srcImg.at<Vec3b>(yi, xi);

Vec3b index01 = srcImg.at<Vec3b>(yi, xi + 1);

Vec3b index10 = srcImg.at<Vec3b>(yi + 1, xi);

Vec3b index11 = srcImg.at<Vec3b>(yi + 1, xi + 1);

dstImg.at<Vec3b>(j, i)[0] = (1 - xd)\*(1 - yd)\*index00[0] + (1 - xd)\*yd\*index01[0] + xd\*(1 - yd)\*index10[0] + xd\*yd\*index11[0];

dstImg.at<Vec3b>(j, i)[1] = (1 - xd)\*(1 - yd)\*index00[1] + (1 - xd)\*yd\*index01[1] + xd\*(1 - yd)\*index10[1] + xd\*yd\*index11[1];

dstImg.at<Vec3b>(j, i)[2] = (1 - xd)\*(1 - yd)\*index00[2] + (1 - xd)\*yd\*index01[2] + xd\*(1 - yd)\*index10[2] + xd\*yd\*index11[2];

完成啦

作者：温素年  
链接：https://www.jianshu.com/p/92b5619e7b4b  
来源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。