

在影视制作领域，往往会涉及到将低分辨率的图像放大为高分辨率的图像的问题，有时候还会涉及到非正方形像素到正方形像素的调整问题。本章将在图像放大算法，像素宽高比调整方法以及图像序列放大处理技术等三方面的问题与读者进行探讨。

## 图像放大方法概述

□ 中国新闻社影视部 易珺

### 一、图像放大算法

图像放大有许多算法，其关键在于对未知像素使用何种插值方式。以下我们将具体分析几种常见的算法，然后从放大后的图像是否存在色彩失真，图像的细节是否得到较好的保存，放大过程所需时间是否分配合理等多方面来比较它们的优劣。

当把一个小图像放大的时候，比如放大 400%，我们可以首先依据原来的相邻 4 个像素点的色彩值如图 1，按照放大倍数找到新的 ABCD 像素点的位置并进行对应的填充，但是它们之间存在的大量的像素点，比如图 2p 点的色彩值却是不可知的，需要进行

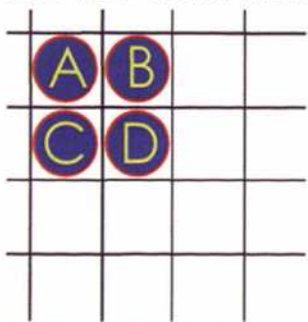


图1 原始图像的相邻4个像素点分布图

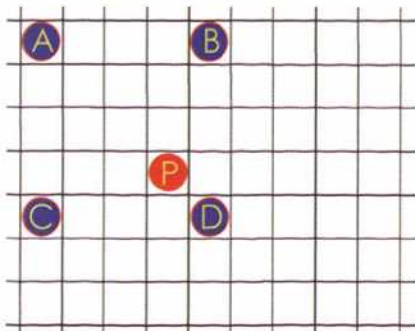


图2 图像放大4倍后已知像素分布图

估算。

#### 1. 最邻近点插值算法 (Nearest Neighbor)

最邻近点插值算法是最简单也是速度最快的一种算法，其做法是放大后未知的像素点P，其位置换算到原始影像上，与原始的邻近的4周像素点A、B、C、D做比较，令P点的像素值等于最靠近的邻近点像素值即可。如上图中的P点，由于最接近D点，所以就直接取  $P=D$ 。

这种方法会带来明显的失真。在A、B中点处的像素值会突然出现一个跳跃，这就是出现马赛克和锯齿等明显走样的原因。最邻近插值法唯一的优点就是速度快。

#### 2. 双线性插值算法 (Bilinear Interpolation)

其做法是：放大后未知的像素点P，其位置换算到原始影像上，计算的四像素点A、B、C、D对P点的影响（越靠近P点取值越大，表明影响也

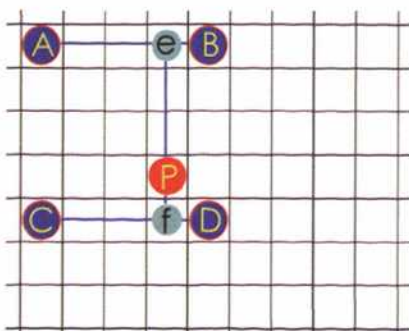


图3 双线性插值算法示意图

越大），其示意图如图3。

其具体的算法分三步，第一步插值计算出AB两点对P点的影响得到e点的值，示意图如图4。

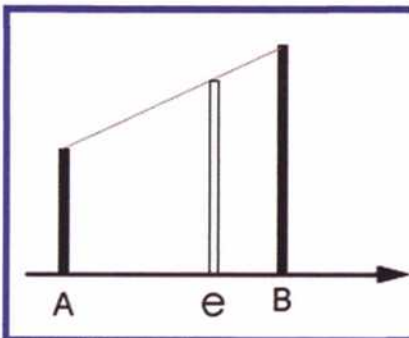


图4 线性插值算法求值示意图

对线性插值的理解是这样的，对于AB两像素点之间的其它像素点的色彩值，认定为直线变化的，要求e点处的值，只需要找到对应位置直线上的点即可。换句话说，AB间任意一点的值只跟A、B有关。

第二步，插值计算出CD两点对P点的影响得到f点的值。

第三步，插值计算出ef两点对P点的影响值。

双线性插值算法由于插值的结果是连续的，所以视觉上会比最邻近点插值算法要好一些，不过运算速度稍微要慢一点，如果讲究速度，是一个不错的折衷。示意图如图5。

#### 3. 双立方插值算法 (Bicubic Interpolation)

双立方插值算法与双线性插值算



法类似，对于放大后未知的像素点P，将其影响的范围扩大到邻近的16个像素点，依据对P点的远近影响进行插值计算，因P点的像素值信息来自16个邻近点，所以可得到较细致的影像，不过速度比较慢。如图6。

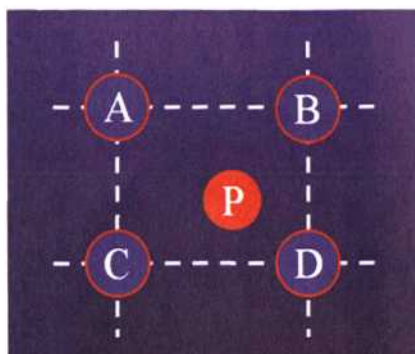


图5 双线性插值算法4个邻近点影响未知点信息示意图

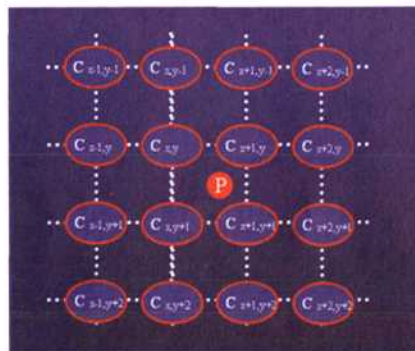


图6 双立方插值算法16个邻近点影响未知点信息示意图

不过双立方插值算法与双线性插值算法的本质区别不仅在于扩大了影响点的范围，还采用高级的插值算法，如图7所示。

要计算A、B两点之间e点的值，需要利用A、B周围A-1、A、B、B+1四个点的像素值，通过某种非线性的计算，得到光滑的曲线，从而算出e点的值来。

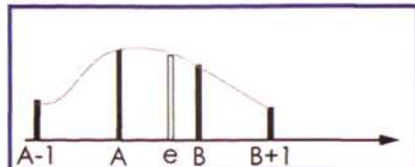


图7 非线性插值算法求值示意图

所谓“双”或者叫“二次”的意思就是在计算了横向插值影响的基础上，把上述运算扩展到二维空间，再

计算纵向插值影响的意思。

双立方插值算法能够得到相对清晰的画面质量，不过计算量也变大。该算法在现在的众多图像处理软件中最为常用，比如Photoshop、After Effects、Avid、Final Cut Pro等。

为了得到更好的图像质量，在以上的基础上，许多新的算法不断涌现，它们使用了更加复杂的改进的插值方式。譬如B样条(B-Spline)、米切尔(Mitchell)等插值算法，它们的目的是使插值的曲线显得更平滑，图像边缘的表现更加完美。

#### 4. 自适应样条插值及其增强技术(S-Spline & S-Spline XL)

与上述经典的插值方法最大的区别在于，S-Spline采用了一种自适应技术，那些传统的方法总是依据周围的像素点来求未知点的色彩值，也就是说需要求解的色彩值仅仅依靠该像素点在图像中的位置，而非实际的图像的像素信息，而自适应样条算法还会考虑实际图像的像素信息。实验表明，经过S-Spline算法得到的图像效果要优于双立方插值算法。

现在S-Spline算法又出现了增强版S-Spline XL，新版本的S-Spline XL算法较S-Spline而言画面的锐度得到进一步增强，物体的轮廓更加清晰，边缘的锯齿现象大大减弱，图像感受更加自然。

## 二、图像放大算法实际测试

现在我们来对以上几种插值算法进行实际测试，看看它们的效果。首先打开原始的图像bird.jpg。这是一个分辨率为360×270的图像，如图8所示。

我们采用BenVista公司著名的PhotoZoom Pro 2软件，在pc电脑上进行测试，电脑的CPU配置为Intel



图8 原始图像



图9 最近点插值 Nearest Neighbour



图10 双线性插值 Bilinear



图11 双立方插值 Cubic



图12 自适应样条插值 S-Spline



图13 自适应样条增强性插值 S-Spline XL



Core 2 T5500 1.66GHz, 分别采用以上 5 种插值方法把原始图像放大 8 倍得到 5 个  $2,880 \times 2,160$  的图像。

对比以上各图并结合实验数据, 我把效果参数列举如下表。

插值类型	主观感受	图像轮廓	总体评价	处理耗时
最临近点插值 Nearest_Neighbour	马赛克现象严重	不清晰	最差	5 秒
双线性插值 Bilinear	图像模糊, 不锐利	边缘不清晰, 有锯齿现象	差	6 秒
双立方插值 Bicubic	图像较模糊, 较锐利	锯齿现象有所改善	折中	8 秒
自适应样条插值 S-Spline	图像相对清晰, 锐利	边缘变得清晰, 锯齿现象消失	好	18 秒
自适应样条增强 S-Spline_XL	图像清晰, 锐利	边缘锐利, 清晰	最好	20 秒

显然, 为了得到最好的图像放大质量, 在可选条件下, 我们应该使用自适应样条插值增强 (S-Spline\_XL) 技术来放大图像。

### 三、像素宽高比调整

现在我们就以 35mm 胶片拷贝的 4K 等效分辨率  $4,096 \times 3,112$  为标准, 使用自适应样条插值增强 (S-Spline\_XL) 技术, 对 SD-NTSC 制式  $720 \times 486$  ( $720 \times 480$ ) 分辨率的图像进行放大。同时来说明如何调整非正方形像素到正方形像素的调整方法。

以 NTSC\_720 $\times$ 486 的图像为例, 它与 4k 胶片等效分辨率  $4,096 \times 3,112$  而言, 存在显示比例不同的问题, 为了最大化保留画面的信息, 我们以信箱模式为例进行调整, 同时我们还要考虑 NTSC 制式像素宽高比变化的问题。

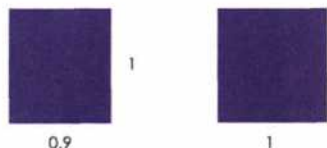


图 14 NTSC 长方形像素示意图

图 15 方形像素示意图



图 16 方形像素显示模式下 NTSC 图像纵向被压缩示意图

我们知道 NTSC 图像的像素的宽

高比并不是 1:1 的, 而是 0.9:1。在方形像素显示模式下, 调用 NTSC 图像时候, 我们发现图像会发生变形, 即感觉画面被压扁了。如图 14、15 所示。

现在我们从 BenVista PhotoZoom Pro 2.3.2 软件中, 先调整像素宽高比变形问题, 再调整画面放大参数。

打开 BenVista PhotoZoom Pro 2.3.2 软件, 它不仅支持单幅图像的放大处理, 还可以批处理图像序列帧, 此版本支持 tif.png.tga.bmp 等图像格式, 在使用的时候我们需要生成软件所支持的格式, 一般用 tif 或 png 格式。

1) 关闭 maintain aspect ratio, 然后预调整新生成的图像宽度为  $720 \times 0.9=648$ , 校正因画面像素宽高比变化导致的画面变形现象, 参数调整如图 17 所示。

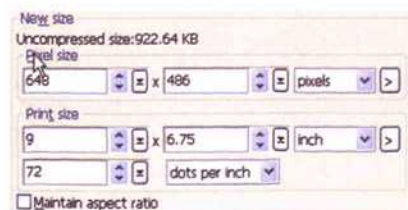


图 17 预调整参数示意图

2) 现在调整新生成的图像宽度值为 4,096, 调整高度值为  $486 \times (4,096/648)=3072$ 。即等比例放大预

生成的图像到 35mm 胶片的等效分辨率大小, 随后定义放大的插值方式为 S-Spline\_XL 即可。

不过由于画幅比例问题, 当使用 35mm 胶片进行记录时, 由于其 4K 等效分辨率为  $4,096 \times 3,112$ , 所以胶片的上下边将留出一定的黑边。

### 四、图像放大批处理

通过以上的分析, 我们了解了使用 BenVista PhotoZoom Pro 2.3.2 软件放大单帧图像的方法, 再结合它的批处理技术, 就可以实现对图像序列的放大了。

在其工具栏中直接启动 new Batch 选项, 然后点击 Add Images 按钮添加图像序列帧, 然后选中所有的帧, 依据源格式以及所需要生成的图像尺寸, 结合上面的分析数据填充放大的后的图像尺寸和插值运算模式, 并制定生成路径后点击运行, 便可以实现图像序列的批处理。



图 18 添加图像序列示意图

经过一定时间的处理, 我们便能得到大尺寸图像序列帧画面了。■

作者: [易珺](#)  
作者单位: [中国新司社影视部](#)  
刊名: [电视字幕·特技与动画](#)  
英文刊名: [TELEVISION CAPTION EFFECTS ANIMATION](#)  
年, 卷(期): [2008, 14\(7\)](#)

## 本文读者也读过(10条)

1. [陆干忠](#), [滕奇志](#), [何小海](#), [吴小强](#) [基于小波变换的图象放大算法的研究](#)[会议论文]-2000
2. [苏丰](#) [图像放大失真的解决方法](#)[期刊论文]-[新课程学习\(学术教育\)](#)2009(12)
3. [陈光](#), [CHEN Guang](#) [数字图像放大算法的研究](#)[期刊论文]-[信息技术](#)2009(6)
4. [吴静](#), [朱宁](#), [WU Jing](#), [ZHU Ning](#) [一种图像放大的离散算法](#)[期刊论文]-[苏州大学学报\(自然科学版\)](#)2005, 21(1)
5. [杨爱萍](#), [王成优](#), [何凯](#), [YANG Ai-ping](#), [WANG Cheng-you](#), [HE Kai](#) [简单有效的小波域图像插值放大算法](#)[期刊论文]-[计算机工程与应用](#)2009, 45(4)
6. [杨云峰](#), [胡金燕](#), [YANG Yun-feng](#), [HU Jin-yan](#) [基于小波变换的图像插值算法](#)[期刊论文]-[大庆石油学院学报](#)2006, 30(4)
7. [崔小球](#) [图像放大精灵—S-Spline](#)[期刊论文]-[网络与信息](#)2002, 16(7)
8. [孙培建](#), [张彩明](#), [高剑](#) [基于加权平均插值曲面的轮廓保持图像放大方法](#)[会议论文]-2005
9. [马天骏](#) [图像放大技术研究](#)[学位论文]2004
10. [石澄贤](#), [吴建成](#), [夏德深](#), [Shi Chenxian](#), [Wu Jiancheng](#), [Xie Deshen](#) [各向异性扩散方程和一种图像放大方法](#)[期刊论文]-[南京大学学报\(数学半年刊\)](#)2005, 22(1)

本文链接: [http://d.wanfangdata.com.cn/Periodical\\_dszmtjydh200807009.aspx](http://d.wanfangdata.com.cn/Periodical_dszmtjydh200807009.aspx)