

## 数字图像插值算法比较研究

盛晓艳, 龚超

(南京林业大学, 江苏 南京 210037)

摘要: 图像插值是图像处理中最基本的技术之一, 得到了广泛研究和应用。文中详细地阐述了常用的4种图像插值算法: 最近邻插值、双线性插值、双三次插值、Lanczos插值。从图像插值的一般原理开始, 对4种算法的数学公式进行了推导, 并给出了部分算法的实现代码。然后, 通过在Visual Studio 2015 + OpenCV 3.2平台上的实验, 分别对4种算法插值后的图像进行主客观两方面的评价, 并对算法运行效率做了比较。

关键词: 图像插值; 兰索斯插值; 原理; 公式

中图分类号: TP317 文献标识码: A

文章编号: 1009-3044(2019)08-0153-04

DOI: 10.14004/j.cnki.ckt.2019.0776

Research on Digital Image Interpolation Algorithm

SHENG Xiao-yan, GONG Chao

(Nanjing Forestry University, Nanjing 210037, China)

Abstract: Image interpolation is one of the most basic technologies in image processing, and has been widely studied and applied. This paper detailly explains 4 commonly used image interpolation algorithms, including nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, Lanczos interpolation. Starting from the general principle of image interpolation, the mathematical formulas of the 4 algorithms are derived, and the implementation codes of some algorithms are given. Then, through the experiments on the platform of Visual Studio 2015 + OpenCV 3.2, the assessment of the subjective and objective two aspects of the interpolated images and the comparison of the operating efficiency are obtained by the 4 algorithms.

Key words: image interpolation; Lanczos interpolation; principle; formula



开放科学(资源服务)标识码(OSID):

图像缩放广泛应用于军事、航空、医学、手持设备和多媒体设备等领域的图像处理与显示。比如日常生活中, 当用户通过4K电视机观看节目时, 由于缺乏4K节目源, 就需要将低分辨率的图像转换为高分辨率的图像, 这就需要用到图像插值技术。图像缩放的基本原理就是根据原图像的像素值通过一定的规则计算得到目标图像的像素值。在图像的缩放过程中, 最重要的就是确定下面两个问题: 一是计算目标图像中的每一个像素值时, 应该选取原图像中哪些像素值; 二是这些选取出的像素值在计算目标像素时权重如何确定。

图像插值算法的一般思路<sup>[1]</sup>是这样的。原图像的像素位于整数坐标表示的网格上, 比如, 我们说某像素点的坐标是(10, 17)。当把这个整数的坐标映射到新的图像后会出现两个问题: 一是原像素的整数坐标被映射到目标图像的小数坐标, 这样的话, 必须求整到最近的整数坐标。二是某些目标图像的位置没有从原像素的位置映射坐标(比如, 在高度方向放大原图像两倍, 则目标图像每隔一行就是空的)。这些问题一般被称作前向投影问题(forward projection)。为了解决这些问题, 我们采用后向解决的思路: 我们逐个遍历目标图像的每一像素点并询问“原图像上的哪些像素点需要映射到这个目标像素点?”而

这些原像素点几乎总是位于小数的位置。所以必须插值原像素点以求出目标像素点的值。

### 1 常用插值算法

目前比较常用的插值算法有这么几种: 最邻近插值, 双线性插值, 双三次插值, Lanczos插值等等, 下面我们就逐一阐述这些算法。

#### 1.1 最邻近插值

最邻近插值算法是最简单的一种插值方法。思路是: 根据目标像素的坐标计算出在原图像中的坐标(待采样点的坐标)。用原图像中最靠近待采样点的像素点的颜色值表示目标像素值。设 $(i+u, j+v)$  ( $i, j$  为正整数,  $u, v$  为大于零小于1的小数, 下同)是待采样点的坐标, 则求目标像素的值  $f(i+u, j+v)$  如图1所示。

收稿日期: 2019-01-05

作者简介: 盛晓艳(1978—), 男, 浙江金华人, 实验师, 主要研究方向为图像处理, 计算机视觉; 龚超(1980—), 男, 湖北蕲春人, 工程师, 主要研究方向为图像检索, 信息检索。

本栏目责任编辑: 唐一东

人工智能及识别技术 153

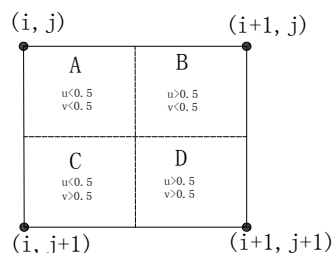


图1 最近邻插值示意图

如果 $(i+u, j+v)$ 落在A区,即 $u < 0.5, v < 0.5$ ,则将左上角像素点 $(i, j)$ 的值赋给 $f(i+u, j+v)$ ,同理,落在B区则赋予右上角的像素点 $(i+1, j)$ 的值,落在C区则赋予左下角像素点 $(i, j+1)$ 的值,落在D区则赋予右下角像素点 $(i+1, j+1)$ 的值。

下面给出了算法的实现代码。matSrc表示原图像,matDst表示缩放后的目标图像。scale\_x, scale\_y分别表示x,y方向的缩放因子。matDst.cols, matDst.rows分别表示目标图像的列数和行数。sx, sy分别表示目标像素点的坐标 $(i, j)$ 对应于与原图像中的坐标。cv::saturate\_cast<int>(i \* scale\_x)这个模板函数在这里相当于求 $i * scale_x$ 这个双精度值的整形值。

```
cv::Mat matSrc, matDst;
matSrc = cv::imread("lena.jpg", 2 | 4);
matDst = cv::Mat(cv::Size(800, 1000), matSrc.type(), cv::Scalar::all(0));
double scale_x = (double)matSrc.cols / matDst.cols;
double scale_y = (double)matSrc.rows / matDst.rows;
for (int i = 0; i < matDst.cols; ++i)
{
    int sx = cv::saturate_cast<int>(i * scale_x);
    for (int j = 0; j < matDst.rows; ++j)
    {
        int sy = cv::saturate_cast<int>(j * scale_y);
        matDst.at<Vec3b>(j, i) = matSrc.at<Vec3b>(sy, sx);
    }
}
cv::imwrite("nearest_1.jpg", matDst);
```

这个算法的优点是计算简单方便,缺点是图像容易出现锯齿。

## 1.2 双线性插值

一维线性插值问题即已知 $(x_0, y_0), (x_1, y_1)$ 情况下,求解 $P(x) = ax + a_0$ 使得 $P(x_i) = y_i$ ,求解思路如图2所示。

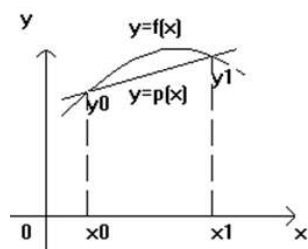


图2 一维线性插值图解

根据点斜式可得:

$$P(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 \quad (1)$$

若用 $f(x_0), f(x_1)$ 来表示 $y_0, y_1$ ,公式(1)可改写为:

$$P(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (2)$$

$P(x)$ 也称为拉格朗日差值多项式<sup>[2]</sup>。在理解了一维的线性插值后,那么二维的线性插值就容易理解了。也就是双线性插值算法中已知四个像素点的颜色值,求中间某一像素点的颜色值 $f(x, y)$ ,如图3所示。

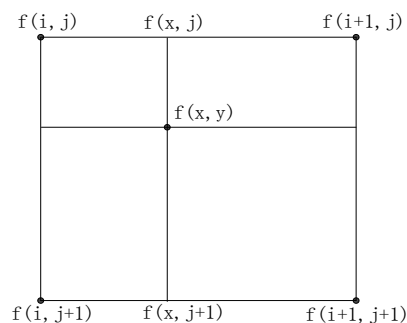


图3 双线性插值示意图

首先在水平方向上运用一维线性插值,公式(2),由 $f(i, j)$ 和 $f(i+1, j)$ 求取 $f(x, j)$ ,由

$f(i, j+1)$ 和 $f(i+1, j+1)$ 求取 $f(x, j+1)$ ,然后在竖直方向运用一维线性插值,由 $f(x, j)$ 和 $f(x, j+1)$ 求取 $f(x, y)$ 。

$$f(x, j) = (i+1-x)f(i, j) + (x-i)f(i+1, j) \quad (3)$$

$$f(x, j+1) = (i+1-x)f(i, j+1) + (x-i)f(i+1, j+1) \quad (4)$$

$$f(x, y) = (j+1-y)f(x, j) + (y-j)f(x, j+1) \quad (5)$$

把(3)和(4)代入(5)得:

$$f(x, y) = (j+1-y)((i+1-x)f(i, j) + (x-i)f(i+1, j)) + (y-j)((i+1-x)f(i, j+1) + (x-i)f(i+1, j+1)) \quad (6)$$

令 $x=i+p, y=j+q$ 得:

$$f(i+p, j+q) = (1-p)(1-q)f(i, j) + p(1-q)f(i+1, j) + (1-p)qf(i, j+1) + pqf(i+1, j+1) \quad (7)$$

公式(7)即为双线性插值算法中计算目标图像中坐标为 $(i, j)$ 点的颜色值函数。

## 1.3 双三次插值<sup>[3-4]</sup>

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个点的颜色值作三次插值,不仅考虑到4个直接相邻点的颜色值影响,而且考虑到各邻点间颜色值变化率的影响。待采样点 $(x, y)$ 的颜色值由其周围16个点的颜色值加权内插得到,如图4所示。

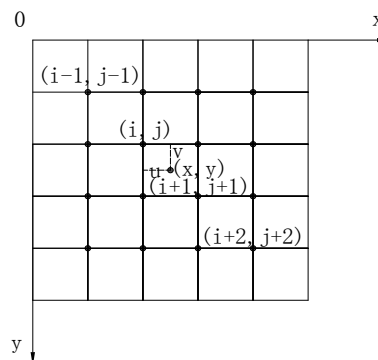


图4 双三次插值示意图

这种算法需要选取插值基函数(16个点的权值)来拟合数据,最常用的插值基函数<sup>[5]</sup>如图5所示。

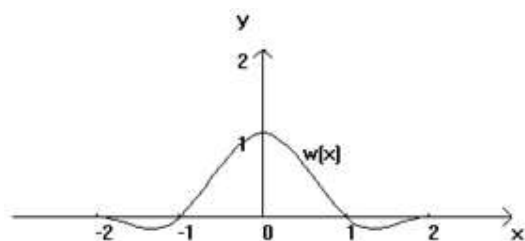


图5 双三次插值基函数

解析式如下:

$$w(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 < |x| < 2 \\ 0 & \text{其它} \end{cases} \quad (8)$$

其中,  $a$ 取-1。则目标像素点的颜色值由下式计算:

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 f(x_i, y_j) w(x - x_i) w(y - y_j) \quad (9)$$

令  $x=i+u, y=j+v$  则

$$f(x, y) = f(i + u, j + v) = ABC \quad (10)$$

其中

$$A = [w(1+v) \ w(v) \ w(1-v) \ w(2-v)]$$

$$B = \begin{bmatrix} f(i-1, j-1) & f(i, j-1) & f(i+1, j-1) & f(i+2, j-1) \\ f(i-1, j) & f(i, j) & f(i+1, j) & f(i+2, j) \\ f(i-1, j+1) & f(i, j+1) & f(i+1, j+1) & f(i+2, j+1) \\ f(i-1, j+2) & f(i, j+2) & f(i+1, j+2) & f(i+2, j+2) \end{bmatrix}$$

$$C = \begin{bmatrix} w(1+u) \\ w(u) \\ w(1-u) \\ w(2-u) \end{bmatrix}$$

公式(9)或(10)即为双三次插值算法中计算目标图像中坐标为  $(i, j)$  点的颜色值函数。

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。

#### 1.4 Lanczos 插值算法

Lanczos 插值算法是取待采样点周围  $8 \times 8$  个,即 64 个像素点的颜色值来表示目标像素点的颜色值。我们首先引出一维情况下的 Lanczos 插值算法。已知  $2a$  个点的函数值,求  $x$  点的函数值。则插值基函数<sup>[6]</sup>(也称为 Lanczos 核函数)如图6。

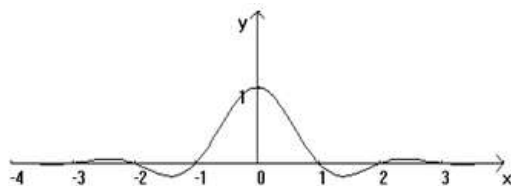


图6 Lanczos 核函数  $a=4$

解析式为

$$L(x) = \begin{cases} 1 & x = 0 \\ \frac{a \sin(\pi x) \sin(\frac{\pi x}{a})}{\pi^2 x^2} & 0 < |x| < a \\ 0 & \text{其它} \end{cases} \quad (11)$$

那么目标像素点的颜色值由下式计算:

$$f(x, y) = \sum_{i=0}^7 \sum_{j=0}^7 f(x_i, y_j) L(x - x_i) L(y - y_j) \quad (12)$$

令  $x=i+u, y=j+v$  则

$$f(x, y) = f(i + u, j + v) = ABC \quad (13)$$

其中

$$A = [L(3+v) \ L(2+v) \ L(1+v) \ L(v) \ L(1-v) \ L(2-v) \ L(3-v) \ L(4-v)]$$

$$B = \begin{pmatrix} f(i-3, j-3) & \cdots & f(i+4, j-3) \\ \vdots & \ddots & \vdots \\ f(i-3, j+4) & \cdots & f(i+4, j+4) \end{pmatrix}$$

$$C = \begin{bmatrix} L(3+u) \\ L(2+u) \\ L(1+u) \\ L(u) \\ L(1-u) \\ L(2-u) \\ L(3-u) \\ L(4-u) \end{bmatrix}$$

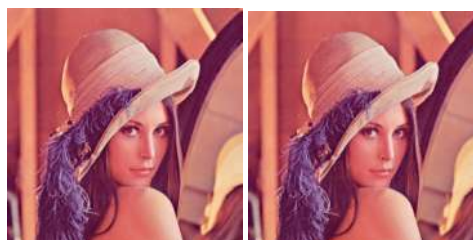
公式(12)或(13)即为双三次插值算法中计算目标图像中坐标为  $(i, j)$  点的颜色值函数。

## 2 实验结果及评价

目前,评价图像插值质量优劣的方式主要有两种标准,一种是主观评价<sup>[7-8]</sup>,即设计实验,由观测者对图像质量进行评价;另一种是客观评价<sup>[9-11]</sup>,采用算法对图像质量进行评价。我们首先取标准测试图像 lena.tif(512x512)的奇数行奇数列组成新图像 quarter.tif(256x256),然后在 quarter.tif 上分别运用上文提到的 4 种插值算法,得到与原图一样大小的图像 nearest.tif(512x512),bilinear.tif(512x512),bicubic.tif(512x512), lanczos.tif(512x512)。我们的实验是运行在 Visual Studio 2015 和 OpenCV 3.2 的环境下,CPU 是 Intel Xeon E5620 @2.40GHz,内存 8GB。

### 2.1 主观评价

主观评价<sup>[7-8]</sup>指通过人眼对于图像的观察,从视觉效果上做出评价。



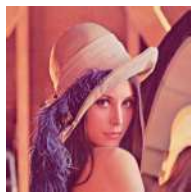
原图

最近邻插值

双线性插值

双三次插值





Lanczos 插值

图7 各算法对“lena.tif”图像的插值效果比较

由于排版的需要,经过缩小后,图7中各幅图像的差别较难发现。从未经缩小的实验结果原图中能看出,除了极小的四周边缘部分之外,lanczos算法要优于bicubic算法,bicubic算法要优于bilinear算法,bilinear算法要优于nearest算法。

主观评价方法会受到观测者专业背景、心理和动机等主观因素的影响,研究人员又提出客观评价方法,即通过一定的算法计算得出一个数值,用数值的大小表示图像的插值质量,这里列出两种常用的客观评价方法。

## 2.2 客观评价

客观评价<sup>[9-11]</sup>指通过插值运算得到的图像与原图像的像素,按某一规则计算得到指标值的大小来衡量图像质量的好坏。由于不同算法得到的指标值往往不同,并且以指标值的形式来呈现图像质量的好坏比主观的评价更具有可比性,因此客观的评价方法逐渐成了该领域评价插值算法优劣的主流。但是,单一的规则往往存在一定的缺陷,不能全面地概括图像的好坏。为了解决这个问题,学者们提出了多种不同的计算规则,从不同的角度去探讨图像质量的评价标准。目前较为流行的评价标准主要有均方误差法<sup>[9]</sup>、峰值信噪比法<sup>[10]</sup>等等。

(1) 均方误差 (Mean square error, MSE) 是指原图像和插值图像各像素点颜色值差异的平均数。MSE 的计算公式为:

$$MSE = \frac{\sum_{i=1}^m \sum_{j=1}^n |K(i,j) - I(i,j)|^2}{m \times n} \quad (14)$$

式中:K和I分别表示插值后的图像与原图像,m,n表示图像的行、列数。MSE的值越小,表明R和I的差异越小,即插值算法越好。反之数值越高表明两幅图像的差异越大,即插值算法越差。

(2) 峰值信噪比 (Peak signal to noise ratio, PSNR) 是一种比较常用的图像评价指标,计算公式为:

$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right) \quad (15)$$

式中:MAX表示图像颜色的最大数值,8位图像取值为255;MSE为图像之间的均方误差。PSNR的单位为dB,数值越大表示两个图像越相似,也就是插值效果越好。普遍基准为30dB,30dB以下的图像劣化较为明显。实验结果显示,PSNR值并不能总与人眼看到的视觉品质一致。

表1 各算法对quarter.tif进行插值得到的图像与原图像lena.tif的客观比较

	Nearest	Bilinear	Bicubic	Lanczos
MSE	42.18	26.67	42.57	45.30
PSNR	31.88	33.87	31.84	31.57

## 2.3 算法效率评价

我们分别在各算法的前后加入cv::getTickCount()和cv::get-

TickCount()这两个函数来获取CPU的运行时间。它们的CPU的时间见表2,单位是秒。

表2 不同算法的时间比较

	Nearest	Bilinear	Bicubic	lanczos
CPU time	0.080	0.705	2.204	7.797

从表2中可以看出,从左到右每个算法所用的CPU时间逐渐增加,这个与每个算法的理论是相吻合的,因为它们分别是1,2×2,4×4,8×8个像素的颜色值来计算目标图像像素的颜色值。

## 3 结束语

本文通过对常用的4种图像插值算法原理和实际代码验证分析,我们发现这4种插值算法中,Lanczos插值和双三次插值的效果最好,但比较耗时间,最近邻插值的效果较差,但速度最快。双线性插值在效果和速度上都介于上面的算法之间,在一般的应用中常被选用。展望未来,作为一个很有实用价值的研究领域,图像插值将会有更多的研究和应用。

## 参考文献:

- [1] Adrian Kaehler, Gary Bradski. Learning OpenCV 3 [M]. O'Reilly,2016: 300-301.
- [2] 张铁,阎家斌.数值分析[M].冶金工业出版社,2007: 143-145.
- [3] Zhou D, Shen X, Dong W. Image zooming using directional cubic convolution interpolation[J].IET Image Processing,2012, 6(6):627-634.
- [4] Hou H S, Andrews H. Cubic splines for image interpolation and digital filtering [J]. IEEE Transactions on Acoustics Speech & Signal Processing,1979,26(6):508-517.
- [5] Robert G. Keys. Cubic convolution interpolation for digital image processing [J]. IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing, 1981,29(6):1153 - 1160.
- [6] Wilhelm Burger, Mark J. Burge. Principles of digital image processing: core algorithms [M]. Springer, 2009: 231 - 232.
- [7] Sheikh H R, Bovik A C. Image information and visual quality [J]. IEEE Transactions on Image Processing, 2006,15(2): 430-444.
- [8] Liu S, Wu L, Gong Y, et al. Overview of image quality assessment[J].Science Paper Online,2011,6(7): 501-506.
- [9] Sakowicz B, Kaminski M, Ritter R, et al. Methods of 3D images quality assessment[C]. IEEE International Conference on Mixed Design of Integrated Circuits & Systems. [S. l.]: IEEE,2015:123-128.
- [10] Sheikh H R, Sabir M F, Bovik A C. A statistical evaluation of recent full reference image quality assessment algorithms[J]. IEEE Transactions on Image Processing, 2006,15(11):3440-3451.
- [11] Wang Z, Li Q. Information content weighting for perceptual image quality assessment [J]. IEEE Transactions on Image Processing, 2011,20(5):1185-1198.

【通联编辑:唐一东】