

学生学号	0121610870910	课程成绩	
------	---------------	------	--

武汉理工大学

学 生 课 程 结 业 论 文

课 程 名 称 计算机图像处理的信息隐藏研究

指导老师姓名 余永升

学 生 姓 名 冯钢果

学生专业班级 软件 1604 班

2017——2018 学年 第 2 学期

关于图像缩放算法介绍

冯钢果

(武汉理工大学 计算机科学与技术学院, 湖北 武汉 430074)

摘要: 文章主要介绍与图像缩放有关的算法, 并通过实验记录数据比较得出所列举算法的效率。首先提出了关于图像缩放 (Image Scaling) 算法的综述和分类, 然后介绍了图像缩放基本原理, 再接着主要介绍了两种算法的实现方法, 最后根据实测量数据比较了这种方法的优劣。结果表明这两种算法都能完成图像缩放, 但效率和质量上大致成反比关系。

关键词: 图像缩放; 插值算法; 图像质量评价

About Image Scaling Algorithm Introduction

Feng Gangguo

(Wuhan University of Technology, Wuhan 430074, China)

Abstract: The article mainly introduces the algorithms related to image scaling, and compares the experimental records with the efficiency of the listed algorithms. Firstly, it presents the overview and classification of Image Scaling algorithm, then introduces the basic principles of image scaling. Then it mainly introduces the implementation methods of the two algorithms. Finally, the advantages and disadvantages of this method are compared based on the actual measurement data. The results show that both algorithms can complete image scaling, but the efficiency and quality are roughly inversely related.

Key words: image scaling; interpolation algorithm; image quality evaluation

引言

近年来随着计算机科学技术的发展, 计算机图像信息处理已经成为一个重要的分支学科, 图像缩放 (Image Scaling) 技术是计算机图像信息处理应用最广的一个相对成熟的技术, 是近年来图像处理学科比较热门的话题, 其主要目的是通过改变分辨率使图像能够在输出设备上获得更好的显示效果, 并且广泛应用于军事、航空、医学、通讯、气象、遥感、动画制作和电影合成等领域。

在计算机图形和数字成像中, 图像缩放是指对数字图像的大小进行调整的过程。

从数字信号处理的角度来看, 光栅图形的缩放是采样率转换的一个二维示例, 即离散信号从采样率转换为另一个采样率; 从奈奎斯特采样定理的角度来看, 图像缩放可以解释为图像重采样或图像重建的一种形式, 因此, 图像缩放又称为图像重采样或图像分辨率转换。

图像缩放包括图像放大和图像缩小两部分, 图像放大即是把低分辨率图像转换为高分辨率图像的过程, 对图像进行升采样; 图像缩小即是把高分辨率图像转换为低分辨率图像的过程, 对图像进行降采样。

图像插值就是利用已知邻近像素点的灰度值来产生未知像素点的灰度值或者通过特殊方法从已经存在的像素点获取目标像素点的灰度值, 以便由原始图像再生出具有更高分辨率或更低分辨率的图像。本文先对图像缩放相关原理进行介绍, 然后介绍三种差值方法: 最近邻居插值、双线性插值算法、双立方插值算法, 最后给出三种插值方法的比较和总结。

1. 图像缩放原理

如何进行图像缩放关系到图像信息的取样和量化，因为图像放大可看作过采样，图像缩小可看作欠采样，这两种操作与取样和量化一幅原始图像的关键区别在于放大和收缩适用于数字图像。

图像放大要求执行两步操作：先创立新的像素位置，再对这些像素位置进行灰度值赋值。例如，一个像素为 100×100 的原始图像，要求按 1.5×1.5 倍放大图像，即得到 150×150 的放大图像。从信息学的角度来讲，原始图像的信息量是增加的，但如果按照信息守恒定律可得：系统中储存信息的变化 = 进入系统的信息 - 离开系统的信息，而这些信息的增加量不能无根据的凭空产生，为此我们可以应用一个简单方法来解决：先建立一个 150×150 的栅栏，为了给任何像素位置都灰度赋值，我们可以简单的应用寻找最靠近的像素的灰度值，并把它赋值给原始图像不存在的新像素，即可实现信息的“增加”，而这种方法是最近邻近插值法，下面会给出具体的实现，当然还有很多插值方法。插值主要的目的是实现对放大图像产生与原始图像拟合新的像素位置和灰度值。当这个灰度值和像素位置与原始图像越拟合，其放大效果就越好。

图像缩小与上述方法相似，而像素增加的过程在此更改为行列删除就实现了图像缩小，例如可以每隔一行（一列）删除一行（一列）进行 0.5×0.5 缩小。可以用放大栅格模拟非整数参数收缩，先假想用扩大的栅格在愿图像拟合，做灰度级最近邻近或双线性内插，然后把栅格收缩到规定大小，就可得到收缩后的图像。

2. 图像插值方法

在数值分析中插值算法可用通式 $g(x) = \sum_{k=0}^{n-1} C_k \times h(x - x_k)$ ，其中 $h(x - x_k)$ 为插值基函数， C_k 为第 k 个原函数值。不同的插值函数只是基函数和选取的插值点个数 n 不同。在此介绍最近邻近插值、双线性插值、双立方插值算法。

2.1 最近邻近插值

最近邻近插值（Nearest-neighbor interpolation）是最简单的灰度值插值，并且是图像缩放技术的最简单和最快的实现，也称作零阶插值，就是令变换后像素的灰度值等于距它最近的输入像素的灰度值。

最近邻近插值基函数：

$$W(x) = \begin{cases} 1 & 0 \leq |x| \leq 0.5 \\ 0 & \text{其他} \end{cases}$$

图像缩放的原则是有一个参考图像，并使用此图像作为基础来构建新的缩放图像。根据缩放比例，构建的图像尺寸会更小、更大或相等。以放大图像为例，放大图像时，我们实际上是在原始基本图片中引入空白空间。图 1.1（为了简化问题，可基本看为 4×4 个像素点组成），将尺寸（ $w_1 = 4$, $h_1 = 4$ ）的图像放大为（ $w_2 = 10$, $h_2 = 10$ ），表格 1 可分别为 4×4 像素的灰度值对应列表。

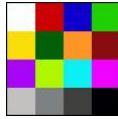


图 1.1

	0	1	2	3
0	(255, 255, 255)	(206, 0, 0)	(10, 0, 206)	(39, 206, 0)
1	(249, 222, 6)	(2, 94, 13)	(254, 155, 38)	(140, 9, 9)
2	(171, 6, 248)	(172, 250, 6)	(62, 40, 249)	(236, 6, 250)
3	(192, 192, 192)	(128, 128, 128)	(64, 64, 64)	(0, 0, 0)

表格 1

这个放大是对原图像进行非整数倍放大，不能简单的将行或列有规律的复制，但可以使用最近邻近插值方法获取新的像素位置和灰度值。最近邻算法基于线性插值。可以先将上面的图像的第一行作为一行。沿线的每个点可以视为线长度的距离的百分比(将每个点除以线的长度，即图像的宽度，4)。原始图像第一行像素灰度值可根据像素点看作一个标尺如图 1.2。

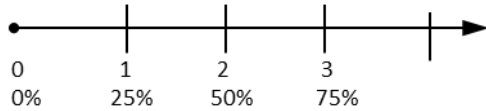


图 1.2

放大后后图像标尺如图 1.3。

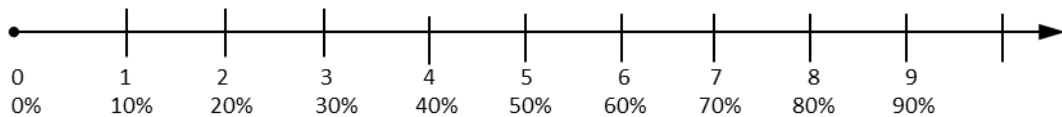


图 1.3

我们可以将变换后图像成为行宽度成为 10 个单位，而原始图像是变换图像的 0.4 倍，因此可以以第一行为例，可以将位置分别乘以 0.4 即可获得新图像的像素对应的灰度值。如下图给出第一行对应值，由于像素坐标是整数，因此我们必须使用最接近的整数像素坐标到此插值的实际值像素。我们通过简单地舍入到最接近的整数来完成此操作。

x (10x10)	x interpolated	x rounded	nearest pixel
0	0.0	0	255,255,255
1	0.4	0	255,255,255
2	0.8	1	206,0,0
3	1.2	1	206,0,0
4	1.6	2	10,0,206
5	2.0	2	10,0,206
6	2.4	2	10,0,206
7	2.8	3	39,206,0
8	3.2	3	39,206,0
9	3.6	4	39,206,0

表 2

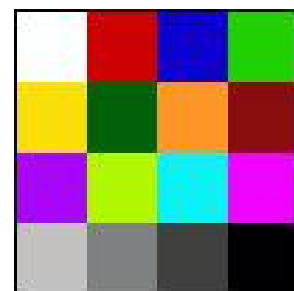


图 1.5

注意，最后一个点（9）将舍入为不存在像素的内插值 4。在这种情况下，最近的像素是前一个像素。必须注意不要访问不存在的像素值。使用高度作为百分比的除数，也沿着 Y 轴使用类似的插值。以下是基本实现方法：（sourceX 和 sourceY 代表原始像素位置，round 为四舍五入函数，targetX 和 targetY 为目标位置，Width 和 Height 代表宽度和高度）

```
sourceX = int( round ( targetX / targetWidth * sourceWidth ) )
sourceY = int( round ( targetY / targetHeight * sourceHeight ) )
```

2.2 双线性插值

除了插值之外，双线性图像缩放与最近邻图像缩放大致相同。而不是复制相邻像素（这通常导致锯齿形图像），基于周围像素的插值技术被用于产生更平滑的缩放。

双线性插值基函数：

$$W(x) = \begin{cases} 1-|x| & 0 \leq |x| \leq 1 \\ 0 & \text{其他} \end{cases}$$

双线性插值算法描述如下：

对于一个目的像素，设置坐标通过反向变换得到的浮点坐标为 $(i+u, j+v)$ （其中 i, j 均为浮点坐标的整数部分， u, v 为浮点坐标的小数部分，是取值 $[0,1]$ 区间的浮点数），则这个像素得值 $g(i+u, j+v)$ 可由原图像中坐标为 (i, j) 、 $(i+1, j)$ 、 $(i, j+1)$ 、 $(i+1, j+1)$ 所对应的周围四个像素的值决定，转换公式如下：

$$g(i+u, j+v) = (1-u) \times (1-v) \times f(i, j) + (1-u) \times v \times f(i, j+1) + u \times (1-v) \times f(i+1, j) + u \times v \times f(i+1, j+1)$$

其中 $f(i, j)$ 表示源图像 (i, j) 处的像素值， $g(i, j)$ 表示输出图像 (i, j) 处的像素值。

比如，在 3×3 的图像变换为 4×4 的图像，现在假如目标图的像素坐标为 $(1, 1)$ ，那么反推得到的对应于源图的坐标是 $(0.75, 0.75)$ ，这其实只是一个概念上的虚拟像素，实际在源图中并不存在这样一个像素，那么目标图的像素 $(1, 1)$ 的取值不能够由这个虚拟像素来决定，而只能由源图的这四个像素共同决定： $(0, 0)$ $(0, 1)$ $(1, 0)$ $(1, 1)$ ，而由于 $(0.75, 0.75)$ 离 $(1, 1)$ 要更近一些，那么 $(1, 1)$ 所起的决定作用更大一些，这从公式 1 中的系数 $u \times v = 0.75 \times 0.75$ 就可以体现出来，而 $(0.75, 0.75)$ 离 $(0, 0)$ 最远，所以 $(0, 0)$ 所起的决定作用就要小一些，因此靠此种方法得到的图像更具有平滑性。

图 2.2 是 80×50 的图 2.1 扩大 2×2 倍的结果：



图 2.1



图 2.2

2.3 双立方插值

双立方插值（Bicubic interpolation）是二维空间中最常用的插值方法，与双线性插值类似。在这种方法中，函数 f 在点 (x, y) 的值可以通过矩形网格中最近的十六个采样点的加权平均得到，在这里需要使用两个多项式插值三次函数，每个方向使用一个。

假设源图像 A 大小为 $m \times n$ ，缩放后的目标图像 B 的大小为 $M \times N$ 。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。

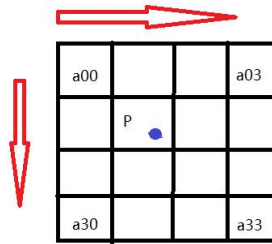


图 3.1

如图 3.1 中， 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。

双立方插值的目的是通过找到一种关系：可以把这 16 个像素对于 P 处像素值得影响因子找出来，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。双立方插值基函数形式如下：

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

其中 a 取 0.5。假设行系数为 $Row(i)$ ，列系数为 $Col(j)$ 。我们以 a_{00} 位置为例：

首先，我们要求出当前像素与 P 点的位置，比如 a_{00} 距离 $P(x+u,y+v)$ 的距离为 $(1+u,1+v)$ 。那么我们可以得到：

$$Row(i)=W(1+u); Col(j)=W(1+v).$$

同理我们可以得到所有行和列对应的系数：

$$Row(0)=W(1+u), Row(1)=W(u), Row(2)=W(1-u), Row(3)=W(2-u);$$

$$Col(0)=W(1+v), Col(1)=W(v), Col(2)=W(1-v), Col(3)=W(2-v);$$

这样我们就分别得到了行和列方向上的系数。然后根据 $k(i,j) = Row(i) * Col(j)$ 得到每个像素 $a(i,j)$ 对应的权值了，最后通过求和公式可以得到目标图片 $B(X,Y)$ 对应的像素值，图 3.2 为 60×60 ，放大为图 3.3 的 120×120 图像。

$$pixelB(X,Y) = pixelA(0,0) * k(0,0) + pixelA(0,1) * k(0,1) + \dots + pixelA(3,3) * k(3,3);$$



图 3.2



图 3.3

3. 差值方法评测

3.1 主观评测

处理效果质量图对比如下：(图 4.1-4.4 分别为原图、最近邻近插值法、双线性插值、双立方插值)



图 4.1



图 4.2



图 4.3



图 4.4

对相同的问题，三种处理方法的效率是：最近邻近插值法 > 双线性插值 > 双立方插值。

从结果来讲：最近邻近插值法 < 双线性插值 < 双立方插值

3.2 客观评测

峰值信噪比（Peak-Signal to Noise Ratio,PSNR）和均方误差（Mean Square Error,MSE）是从统计角度来衡量待评图像的质量优劣，设待评价图像为 F ，参考图像为 R ，它们大小为 $M \times N$ ，则利用 PSNR 和 MSE 的表征图像质量的计算方法为：

$$\text{PSNR} = 10 \lg \frac{255^2}{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |R(i,j) - F(i,j)|^2}$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |R(i,j) - F(i,j)|^2$$

PSNR 与 MSE 都是通过计算待评图像与参考图像之间像素误差的全局大小来衡量图像质量好坏的。PSNR 值越大，表明待评图像与参考图像之间的失真较小，图像质量较好。而 MSE 的值越小，表明图像质量越好。

先分别用三种方法对同一幅图像放大 1.25*1.25 倍，在用之前方法缩小 0.8*0.8 倍，两次变换后即可变为和原始图像大小一致的图片，然后分别记录 PSNR 和 MSE 值并比较。

	最近邻近插值	双线性插值	双立方插值
PSNR	38.3312	42.4264	44.5052
MSE	1.4940	0.8913	0.8119

从 PSNR 和 MSE 值来看，
最近邻近插值法 < 双线性插值 < 双立方插值。

4. 结束语

在图像处理中，进行图像缩放主要是对数字图像信息的重采样，而插值法就是本文主要介绍对象，最近邻近插值法、双线性插值、双立方插值都可对图像进行缩放，但三者效率和效果各有所长，其中最常用的是双线性插值。本文主要结合实际情况对三种插值方法分别进行了介绍，最后并给出了三种插值方法的比较。

参考文献:

- [1] Rafael C.Gonzalez, Richard E.Woods.数字图象处理[M].北京:电子工业出版社,2011. 50-51.
- [2] 王翔.数字图像缩放及图像质量评价关键技术研究[D].杭州:浙江大学,2012. 2-12.
- [3] 张阿珍, 刘政林, 邹雪城, 向祖权.基于双立方插值算法的图像缩放引擎的设计[J].微电子学计算机,2007,24(1):49-51
- [4] 符祥, 郭宝龙.图像插值技术综述[J].多媒体技术,2009,30(1):141-143