

一种新的数字图像放大 $\frac{4}{3}$ 倍快速算法

曾嘉亮¹, 钟似玟²

(1. 汕头职业技术学院 机电工程系, 广东 汕头 515078;

2. 中山大学 计算机应用研究所, 广东 广州 510006)

摘要: 提出一种新的数字图像放大 $\frac{4}{3}$ 倍快速算法。该方法既解决了零阶插值算法的棋盘格效应问题, 又在运算效率上比传统的一阶插值算法提高了 90%。

关键词: 数字图像; 放大; $\frac{4}{3}$ 倍; 快速算法

中图分类号: TN317.4

文献标识码: A

文章编号: 1674-7720(2012)22-0032-03

A new fast algorithm for scaling up image $\frac{4}{3}$ times

Zeng Jialiang¹, Zhong Sifen²

(1. Department of Mechanical & Electrical Engineering, College of Shantou Polytechnic, Shantou 515078, China;

2. Computer Application Institute, Sun Yat-sen University, Guangzhou 510006, China)

Abstract: The paper proposes a new fast algorithm for scaling up image $\frac{4}{3}$ times. It not only removes the checkerboard effect of zero order interpolation, but also gets 90 percent speed improvement than the traditional first order linear interpolation.

Key words: digital image; scaling up; $\frac{4}{3}$ times; fast algorithm

随着移动网络的快速发展, 各种视频应用在手机等移动设备上正变得日益普及。而移动网络具有带宽受限、稳定性差等不足。这决定了其传输的视频流应尽可能小, 以保证应用中传输能够流畅而稳定地进行。

目前, 低端智能手机的屏幕分辨率是 QVGA (320×240), 为节省带宽, 对这类手机传输的视频, 分辨率一般是 240×180^[1], 图像的水平 and 垂直分辨率都是屏幕的 $\frac{3}{4}$, 当需要全屏播放时, 水平、垂直方向均放大 $\frac{4}{3}$ 倍即可。同理, 可以推广到分辨率为 640×480 或 800×480 等大屏幕的智能手机上。

由于手机是资源受限系统, 而数字图像放大算法的复杂度又比较高。因此, 开发出运算效率高、且能兼顾放大后图像质量的 $\frac{4}{3}$ 倍放大算法, 对推动手机端的视频应用具有较强的实用意义。

1 数字图像放大 $\frac{4}{3}$ 倍原理

对数字图像进行 $\frac{4}{3}$ 倍放大, 需要执行两步操作^[2]:

(1) 创立新的像素位置; (2) 对这些新位置赋予灰度值。

《微型机与应用》2012 年 第 31 卷 第 22 期

假设有一幅大小为 3×3 像素的图像, 想把它放大 $\frac{4}{3}$ 倍, 即 4×4 像素。一种较为容易理解的图像放大概念的方法是: 在原始图像上放置虚构的 4×4 个栅格(显然, 栅格的间隔小于 1 个像素); 然后, 采用某种算法将栅格上的点全部赋值; 最后, 将这些栅格扩展到目标图像的大小。这样, 就完成了图像的 $\frac{4}{3}$ 倍放大操作。

原始图像和放大图像在像素位置上的这种关系可以用图 1 来帮助理解。其中, 符号 o 表示原图像像素位置, 符号 x 表示放大图像的像素位置, 符号 ⊗ 表示原像素和放大后的像素在同一位置。

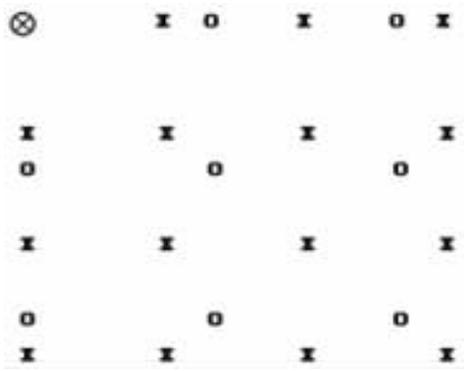
传统的图像放大算法实际上都是上述“某种算法”一词的具体例子, 亦即用于解决给新位置赋予灰度值的问题。

2 传统放大算法应用于 $\frac{4}{3}$ 倍放大

2.1 零阶插值

最简单的放大算法是最近邻插值, 也称零阶插值^[2]。该方法在原图像上寻找一个最靠近的像素, 将其灰度值

欢迎网上投稿 www.pcachina.com 35

图 1 3×3 图像放大 $\frac{4}{3}$ 倍的一般理解

赋给栅格上的新像素。

这种算法的优点是简单快捷,只需取值、赋值,不需要任何运算;但缺点也同样明显——存在严重的棋盘格效应。

2.2 一阶插值

为了提高精度,一般采用一阶放大算法——双线性插值^[2-3]。该算法采用 4 个与新像素最靠近的像素进行双线性内插。令 (x', y') 代表放大图像中一点的坐标,并令 $v(x', y')$ 代表该点被赋予的灰度值。对于双线性插值来说, $v(x', y')$ 由下式给出:

$$v(x', y') = ax' + by' + cx'y' + d \quad (1)$$

其中, a, b, c 和 d 4 个系数由点 (x', y') 的 4 个最邻近点列出的 4 个方程决定。

虽然双线性插值的效果较好,但算法比较复杂,通常算法需要用到浮点运算。即便是对 $\frac{4}{3}$ 倍放大进行专门的优化后,平均生成一个新像素也需要做 2.5 次整数加法和 2.5 次移位操作。

2.3 对一阶快速插值算法的需求

对于手机等资源受限的移动视频终端,在使用 $\frac{4}{3}$ 倍放大算法时,通常不采用零阶插值,因为其明显的棋盘格效应严重影响用户体验;而一阶的双线性插值虽然图像质量令人满意,但运算速度太慢。

为此,本文提出一种新的 $\frac{4}{3}$ 倍放大算法,该算法既具备了一阶算法的插值精度,又满足了资源受限系统对速度的要求。

3 一阶精度的数字图像 $\frac{4}{3}$ 倍放大快速算法

本算法针对 $\frac{4}{3}$ 倍放大中像素间关系的特点而设计,在计算复杂度方面,与零阶的最近邻插值算法最为接近,而在图像质量方面,则与一阶的双线性插值较为接近。下面通过图例,详细介绍本算法的操作步骤。

图 2 说明了本算法的一维操作,即如何将原始图像的 3 点放大为目标图像的 4 点。这里,符号 \circ 表示原图

像素,符号 x 表示放大图像的像素,符号 \otimes 表示原像素和放大后的像素在同一位置。



图 2 3 点变 4 点示意图

图 2 中,水平方向的 0~12 表示理想的图像采样间隔;0、4、8、12 表示实际采样间隔,也就是原始图像的采样间隔;0、3、6、9 表示重新采样的间隔,亦即从原始图像放大而来的目标图像的采样间隔。

令 $v(x)$ 代表原始图像或放大图像的一个像素在一维采样坐标中的位置,并令 $v(x)$ 代表原始图像或放大图像的像素值。图 2 中 4 个新像素的值可以用下列公式获得:

$$v(0) = v(0) \quad (2)$$

$$v(6) = [v(4) + v(8)] \div 2 \quad (3)$$

$$v(3) = [v(0) + v(6)] \div 2 \quad (4)$$

$$v(9) = [v(6) + v(12)] \div 2 \quad (5)$$

对于图像在二维方向上都放大 $\frac{4}{3}$ 倍的情况,只需将式(2)~式(5)先在水平方向上应用一次,然后在垂直方向上再应用一次,即可生成最终的放大图像。由于整数的除 2 操作可以用右移来代替,因此平均算下来,生成一个新像素需要 1.3125 次整数加法和 1.3125 次移位操作,复杂度比双线性插值算法小了近 50%。

式(2)~式(5)是本算法的核心内容,却还不是全部问题的答案。

考虑位于原始图像右边界或下边界处的 3 点如何插值出 4 点。这种情况下,采样坐标 12 (见图 2) 的位置已经超出了图像的范围,其像素值未知。此时,必须对 $v(9)$ 的求法略做调整:

$$v(9) = v(6) \quad (6)$$

当然,如果原始图像的宽度或高度不是 3 的整数倍,则在放大操作前,必须先进行边界扩展,将原图的宽度和高度补齐到 3 的整数倍。

表 1 对比了不同算法产生一个新像素所需要的平均运算量。

表 1 不同算法的运算量对比数据表

生成一个新像素平均需要的	最近邻插值算法	双线性插值算法	本算法
整数加法操作/次	0	2.5	1.3125
移位操作/次	0	2.5	1.3125

表 1 数据表明,本算法的运算效率比同为一阶的双线性插值算法提高了 $(2.5 - 1.3125) / 1.3125 \approx 90\%$;其复杂度介于传统的零阶和一阶算法之间。

4 实验结果

将一幅 320×240 的一阶线性渐变图像(下称 F_320×240,如图 3 所示)作为比照实验中的基准图像,用高精

《微型机与应用》2012 年 第 31 卷 第 22 期

度的 Bicubic^[4] 算法对其进行缩小操作,产生的 240×180 图像(下称 F_{240×180},如图 4 所示)作为插值的源图像。

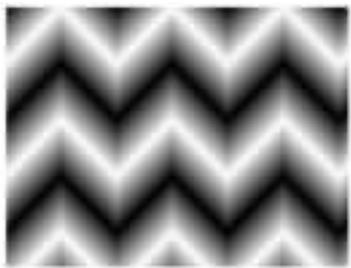


图 3 基准图像 (F_{320×240})

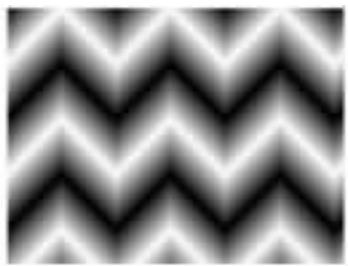


图 4 插值源图像 (F_{240×180})

分别用零阶最近邻插值、本算法和一阶双线性插值对图像 F_{240×180} 进行放大,产生的 3 幅图像对应记为 NE_{240×180}(如图 5 所示)、MY_{240×180}(如图 6 所示)和 BL_{240×180}(如图 7 所示)。从主观评价和客观评价^[5]两个方面对各种放大算法的结果进行对比。

4.1 主观评价

就人眼的主观评价而言,由图 3、图 5、图 6、图 7 不难看出:在参与对比的 3 种算法中,零阶最近邻插值的图像有明显的棋盘格效应,一阶双线性插值的图像保真度最高,而本算法插值出的图像则消除了零阶插值的棋盘格效应,从而在图像质量上更接近双线性插值。

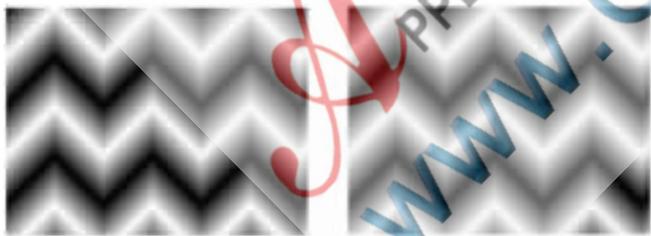


图 5 最近邻插值图像 (NE_{320×240})

图 6 本算法插值图像 (MY_{320×240})

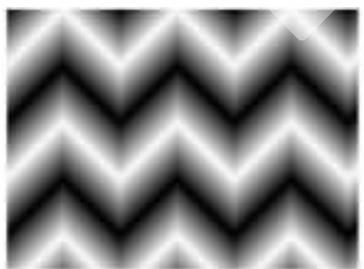


图 7 双线性插值图像 (BI_{320×240})

4.2 客观评价

以图像 F_{320×240} 为基准图像,分别计算零阶插值图像 NE_{320×240}、本算法插值图像 MY_{320×240} 以及一阶双线性插值图像 BI_{320×240} 相对于基准图像的峰值信噪比 PSNR (Peak Signal-to-Noise Ratio)^[5],所得数据如表 2 所示。

表 2 不同算法的 PSNR 对比数据表

	最近邻插值算法	本算法	双线性插值算法
PSNR/dB	36.868	37.805	39.772

由表 2 可知,本算法插值结果的 PSNR,比零阶插值提高了 0.937 dB。

4.3 实验小结

图像质量评价本质上是一种主观评价;而且作为常用的客观评价手段,PSNR 并不能很好地与主观评价相对应^[5]。因此,在本实验中,综合主、客观评价的结果,可以得出结论:新算法插值的结果明显优于零阶算法。

本文提出了一种新的数字图像放大 $\frac{4}{3}$ 倍快速算法,相对于传统的零阶和一阶放大算法,新算法较好地平衡了运算速度和图像质量之间的关系,其计算复杂度与零阶插值接近,而图像质量则接近于一阶的双线性插值。

理论分析及实验结果表明,该方法既消除了零阶插值的棋盘格效应,又在运算效率上比同为一阶的双线性插值提高了 90%,非常适用于在资源受限的系统中进行 $\frac{4}{3}$ 倍放大操作。

参考文献

- [1] 网易公开课下载专区. <http://v.163.com/special/opendownload/>.
- [2] GONZALEZ R C, WOODS R E. 数字图像处理(第二版)[M]. 阮秋琦等,译. 北京:电子工业出版社,2003.
- [3] 章毓晋. 图像工程(上册)图像处理(第二版)[M]. 北京:清华大学出版社,2006.
- [4] BOURKE P. Bicubic interpolation for image scaling[EB/OL].[2012-08-01]. <http://astronomy.swin.edu.au/~pbourke/colour/bicubic>.
- [5] RICHARDSON I E. H.264 and MPEG-4 video compression[M]. Chichester, England: Wiley & Sons Ltd, 2003.

(收稿日期:2012-08-24)

作者简介:

曾嘉亮,男,1976年生,硕士,讲师,主要研究方向:数字图像处理、视频压缩技术及电子信息技术。

钟似玢,男,1950年生,博士,主要研究方向:计算机图形图像处理技术。