

一种新的基于梯度的2*图像插值算法

刘刚¹, 韩建栋²

(1.山东理工大学, 山东 淄博 255049; 2.中北大学, 山西 太原 030051)

摘要:介绍了常见的图像插值放大方法及其优缺点,提出了一种新的2*图像插值放大算法,该算法运算速度快,效果明显,对图像包含信息量大的边缘,保持较好,易于用硬件实现。

关键词:图像插值; 图像边缘; 梯度

中图分类号: TN911.73

文献标识码: A

文章编号: 1001-8891(2006)06-0324-03

A New 2*Image Interpolation Based on Gradient

LIU Gang¹, HAN Jian-dong²

(1. Shandong University of Technology, Shandong Zibo 255049, China)

2. North University of China, Shanxi Taiyuan 030051, China)

Abstract: A new 2* image interpolation based on gradient was proposed as well as common image interpolation methods reviewed. The new image interpolation method had some advantages of fast operation and obvious effect, it kept large information of the image edge and easily realized by hardware.

Key words: image interpolation; image edge; gradient

引言

在很多实时图像处理领域,图像插值是一种常用的算法。尤其随着多媒体技术的发展以及大屏幕的图像显示,图像放大的重要性也越来越明显^[1]。传统的易于硬件实现的图像插值放大算法主要有像素重复放大^[2]、双线性插值放大、双立方插值放大^[3],本文在简要介绍图像插值放大算法的基础上,针对实时图像放大处理系统对实时性要求高以及算法易于硬件实现的要求,在结合线性插值和非线性插值算法原理的基础上,给予图像的梯度信息,提出了一种新的2*图像插值放大算法。

1 像素重复放大算法

像素重复放大是一种最简单也是最原始的图像插值放大,它的基本原理是用原始像素直接对插值点像素用原始像素填充。

2 双线性插值法

双线性插法是指对放大后的图像,对原始像素周围的放大后的新点的像素,用原始图像上原始点与右

边、下边、和右下的像素的平均值代替。也就是输出像素的赋值为 2×2 矩阵所包含的有效点的加权平均值。

3 双立方插值法

双立方插值是对双平方插值的扩展,输出像素的赋值为 4×4 矩阵所包含的有效点的加权平均值。

从以上三种算法可以看出,像素重复放大最简单,但是马赛克现象比较严重,边缘保持不好。双线性插值和双立方插值在边缘上一定程度的克服了重复像素算法带来的马赛克现象,但是由于这两种方法本质上属于“低通滤波器”,对放大图像具有平滑作用,因此带了边缘信息模糊的缺点,高频信息丢失严重,而图像中的高频信息又往往是我们研究的重点。

为了克服双线性插值和双立方插值带来的图像模糊问题,又出现了样条插值、分形插值、小波插值等方法,但是这些算法的复杂度比双线性插值和双立方插值复杂的多,用硬件实现的难度较大。基于上述所述,本文提出了一种新的插值算法,该算法运算简单,效果比双线性插值和双立方插值明显,对边缘保持的比较好。

4 本文算法分析

图像的边缘对应的是图像中的高频信息,也就是

收稿日期: 2005-04-21

作者简介: 刘刚 (1976—), 助教, 主要研究方向: 光电成像, 实时图像处理, 图像通信等。

灰度级变化比较剧烈的地方^[4]，一个边缘的示例如图1所示。

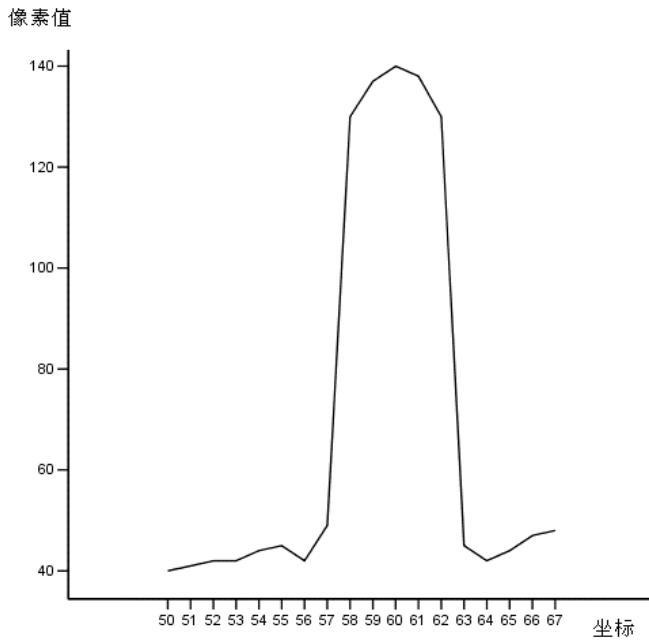


图1 边缘示例 Fig.1 Example of edge

图1的示例边缘比较明显，从图中，我们可以看出对应的横坐标56~58, 62~64的位置对应的是图像的边缘，当我们采用双线性插值或者双立方插值的时候，由于双线性插值和双立方插值的“低通”作用，在灰度变化比较剧烈的地方，插进去了新的值，使得图像边缘变得比较平缓，因此，如果我们在灰度变化比较剧烈的地方（边缘）插值时，插值点尽量靠近低频区域，这样插值点对灰度变化率的影响就会降低，同时较好的保持了图像的边缘。本文算法就是基于这种思想而提出的一种不均匀的插值方法。下面对本文的插值算法进行介绍。

假定图2是一幅图像的一个子图， P_0, P_1, P_2, P_3 分别代表了图示中相应坐标位置的灰度值。同时，我们还将对 P_4, P_5, P_6 这三个相应坐标点的灰度值感兴趣。

P_0	P_1	P_4
P_2	P_3	
P_5		P_6

图2 图像子图示例
Fig.2 Example of sub-image

P_0	In_0	P_1
In_1	In_2	
P_2		P_3

图3 插值后子图示例
Fig.3 Example of interpolated sub-image

在对子图进行插值后，插值点分别为 In_0, In_1, In_2 。进行2*图像插值后，相应插值点如图3。图3中只给出了相对应 P_0 点周围的插值点。

首先进行 In_0 和 In_1 的插值计算。因为边缘在梯度图中，反映的是梯度值比较大的地方，因此，在进行

插值的时候，我们还需要求出 P_0 与 P_1 ， P_0 与 P_2 之间梯度值的绝对值，如式(1)、式(2)所示，式中： Abs 函数代表取绝对值。

$$DG_1 = Abs(P_0 - P_1) \quad (1)$$

$$DG_2 = Abs(P_0 - P_2) \quad (2)$$

In_0 的插值计算公式(3):

$$\left. \begin{array}{l} If (P_0 \leq P_1) \\ \quad In_0 = P_0 + DG_1 * r, \\ Else \\ \quad In_0 = P_1 + DG_1 * r, \end{array} \right\} \quad (3)$$

In_1 的插值计算公式(4):

$$\left. \begin{array}{l} If (P_0 \leq P_2) \\ \quad In_1 = P_0 + DG_2 * r, \\ Else \\ \quad In_1 = P_2 + DG_2 * r, \end{array} \right\} \quad (4)$$

式(3)、式(4)中 r 定义为增值系数，从前面的讨论中，为了保出插值后图像的边缘，在插值时尽量靠近灰度值小的地方，因此，在本文中 r 取0.2，试验中发现，取值在0.15~0.30时，得到的图像比较细腻，且边缘保护良好。从式(3)、式(4)中可以看出，在进行水平插值时，主要利用了水平方向上梯度的信息，在垂直插值时，则主要利用了垂直方向上的梯度信息，利用了靠近像素灰度值小的区域插值的方法，增值系数 r 的存在，使得插值点到边界的变化过渡自然且保持了图像边缘信息，经过插值后，图像边缘还是比较锐利。

由于双线性插值和双立方插值方法放大后，在45°(或225°)和135°(或315°)的边缘方向上，锯齿现象比较明显，因此，本文的插值方法在进行 In_2 插值的时候，充分利用了这两个方向上的梯度信息。 In_2 的计算方法描述如下：

首先，计算 P_0 与 P_3 ， P_3 与 P_4 ， P_3 与 P_5 ， P_3 与 P_6 之间的梯度值的绝对值。分别如式(5)~式(8)。

$$DG_{03} = Abs(P_0 - P_3) \quad (5)$$

$$DG_{36} = Abs(P_3 - P_6) \quad (6)$$

$$DG_{35} = Abs(P_3 - P_5) \quad (7)$$

$$DG_{34} = Abs(P_3 - P_4) \quad (8)$$

然后分别计算在这两个方向上的梯度的绝对值的和。

$$Sum45 = DG_{03} + DG_{36} \quad (9)$$

$$Sum135 = DG_{34} + DG_{35} \quad (10)$$

当 $Sum45$ 的值比 $Sum135$ 大时，则在135°方向上的变化比较缓慢，因此， In_2 的取值主要与135°上的 P_0 和 P_3 有关。反之亦然，为了简化计算方法， In_2

的计算如下:

$$\left. \begin{array}{l} \text{If } (\text{Sum45} > \text{Sum135}) \\ \quad \text{In}_2 = (P_0 + P_3)/2 \\ \text{Else if } (\text{Sum45} < \text{Sum135}) \\ \quad \text{In}_{12} = (P_1 + P_2)/2 \\ \text{Else} \\ \quad \text{公式(12)、(13)、(14)} \end{array} \right\} \quad (11)$$

由公式(11), 当 Sum45 与 Sum135 相等时, 我们从这两个梯度信息上, 已经无法判断插值点 In₂ 的取值, 也就是在 45° 和 135° 上, 我们认为梯度是相等的, 在这种情况下, 我们从垂直和水平两个方向上比较梯度的大小, 从而确定 In₂ 取值, 如公式(12)、公式(13), 我们首先求出在水平和垂直方向上的梯度绝对值的和, 从而确定边缘在那一个方向上。

$$DG_3 = \text{Abs}(P_0 - P_1) + \text{Abs}(P_2 - P_3) \quad (12)$$

$$DG_4 = \text{Abs}(P_0 - P_2) + \text{Abs}(P_1 - P_3) \quad (13)$$

由公式(12)、公式(13), 我们知道, 当 DG₃ 的值大于 DG₄ 时, 则边缘的方向在垂直方向上要明显, 反之, 当两者相等时, 我们不能确定它的方向, 因此为了简化计算, 认为在两个方向上边缘信息是相同的, 此时, 为了简化计算, 以及上面的分析我们得出公式

(14)。

$$\left. \begin{array}{l} \text{If } (DG_3 > DG_4) \\ \quad \text{In}_2 = \text{In}_0 \\ \text{Else if } (DG_3 < DG_4) \\ \quad \text{In}_2 = \text{In}_1 \\ \text{Else} \\ \quad \text{In}_2 = (\text{In}_0 + \text{In}_1)/2 \end{array} \right\} \quad (14)$$

5 结论

该算法从边缘像素区域的灰度值变化的特点进行分析, 利用了相邻像素点的梯度信息, 使插值点的插值尽量减小对图像边缘灰度值变化率的影响, 使得图像边缘比较明显, 锯齿现象较低, 同时在低频区域, 图像的细节也比较细腻, 该算法简单, 因此可以在实时的图像处理系统中利用硬件实现。试验结果如图 4。

本文中, 只对本算法和双立方插值算法的结果进行比较, 原图像是 128×128 的血细胞图像, 从实验图中, 可以看出, 图(c)边缘保持良好, 锯齿现象轻微, 图像低频部分比较细腻, 从试验结果可以看出, 此方法比双立方插值效果要好, 同时, 该算法也易用 HDL 语言在 FPGA 上实现。

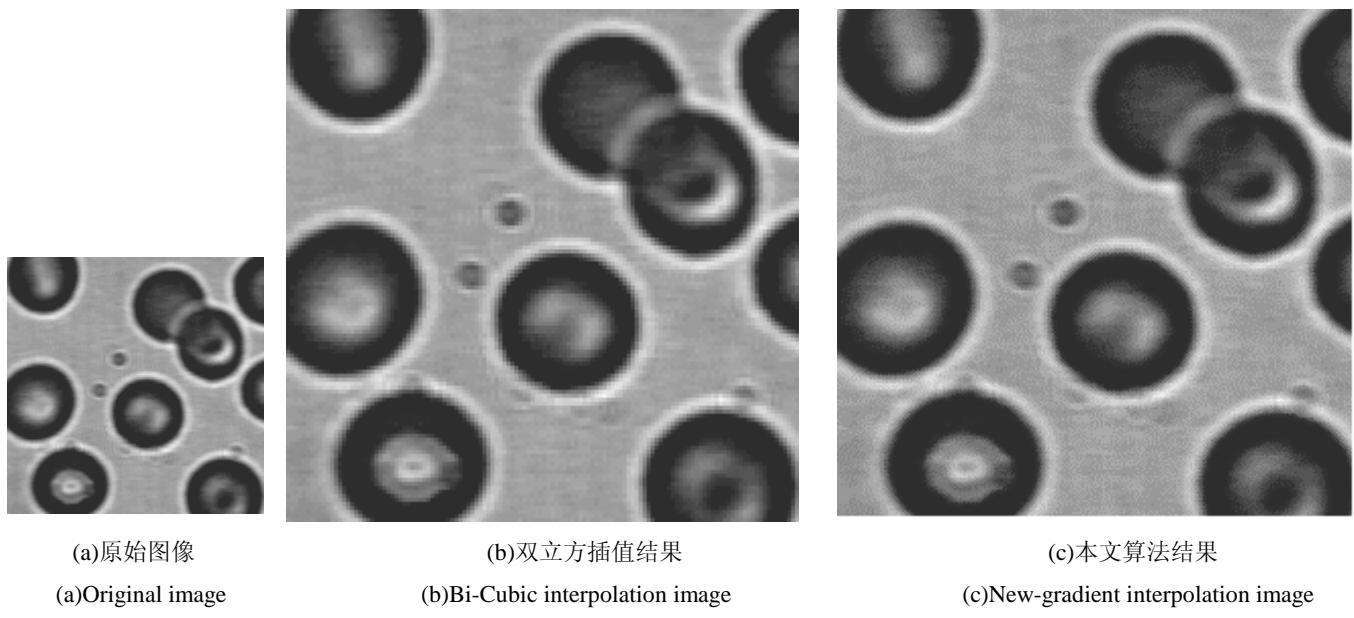


图 4 试验结果

Fig.4 Testing results

参考文献:

- [1] 江铭炎, 李兴江, 袁东风. 2*图像插值放大处理的方法[J]. 山东大学学报(理学版). 2003, 38(3): 79~81.
- [2] 樊雅平, 黄生学. 一种新的基于像素交迭的图像插值方法[J]. 广西师范大学学报. 2003: 72~75
- [3] 韩彦中. 数字图像 FFT 算法及编程的探讨[J]. 红外技术. 2003, 25(3): 27~28.
- [4] 马天骏, 高有行. 基于三次样条的不均匀插值图像放大方法[J]. 电子科技. 2004, (2): 45~47.