

基于 Lanczos 核的实时图像插值算法

郭莹, 李伦, 王鹏

(武汉数字工程研究所, 湖北 武汉 430074)

摘 要: 针对现有彩色图像插值算法的实时性及可靠性不能兼备的问题, 提出了一种高效的实时彩色图像缩放算法, 算法基于 Lanczos 核生成可应用于整个目标图像的核查找表, 并且目标图像所有像素的插值仅涉及定点小数运算, 计算复杂度低; 同时算法采取了优化的光栅扫描方法, 内存需求低、效率高。实验结果表明, 该算法不仅获得了较好的图像质量, 而且计算复杂度远低于经典的图像缩放算法, 能有效实现彩色图像甚至彩色视频的任意倍数实时缩放。

关键词: 实时彩色图像缩放; Lanczos 核; 查找表; 定点小数运算

中图分类号: TP391

文献标识码: A

Real time interpolation algorithm based on Lanczos kernel

GUO Ying, LI Lun, WANG Peng

(Wuhan Digital Engineering Institute, Wuhan 430074, China)

Abstract: In order to improve the timeliness and reliability of color image scaling algorithms, a high effective real-time color image scaling algorithm was proposed, which first generated the kernel look up table that could be used for whole target image based on Lanczos kernel, and interpolation of all target image pixels had only fixed point operation which had low computational complexity. Meanwhile, an optimized method for rasterizing pixels of target image was proposed, which needed little memory space but embraced high efficiency of memory utilization. The experiment results show that the proposed algorithm achieves good image quality, and the computational complexity is far more less than classic image scaling algorithms. So the proposed algorithm can effectively implement real-time resizing of color images and color videos in any scaling factors.

Key words: real-time color image resizing, Lanczos kernel, look up table, fixed point operation

1 引言

实时彩色图像缩放广泛应用于军事、航空、医学、消费电子、手持设备和多媒体设备等领域的图像处理与显示中。经典的线性插值算法包括最近邻法、双线性插值^[1]、双三次插值^[2]算法等。最近邻法复杂度最低, 处理效率最高, 但其缩放后的图像锯齿效应最严重; 双线性插值是处理效率和图像质量的合理折中, 但它生成的图像质量偏低, 并且其每一个目标像素点的生成均需要重新计算插值权重, 效率偏低; 双三次插值生成的图像质量较佳, 但其算法复杂度偏高, 不适用于实时图像缩放。为

了提高图像质量及运算效率, 出现了一系列改进插值算法。文献[3]算法简单高效, 可以实现图像的实时缩放, 但其算法会出现锯齿效应、纹理细节等边缘特征丢失的缺点, 导致图像变模糊。文献[4]利用查找表技术提高了处理效率, 但其提出的图像放大与缩小算法不一致, 导致硬件需要实现 2 套独立的逻辑, 且缓存偏大, 不利于硬件实现。文献[5,6]生成的图像质量略高于双线性插值, 但其算法涉及色彩空间转换, 并且插值权重的计算稍显复杂, 很难实现实时图像缩放。文献[7~9]提出了基于区域的冲击滤波的图像增强算法, 对于不同图像区域采用不同的权重系数处理, 该方法可以取得很好的图像质

收稿日期: 2016-08-10; 修回日期: 2017-03-31

基金项目: 船舶基金资助项目 (No.11J3.7.2)

Foundation Item: The Shipping Foundation of China (No.11J3.7.2)

量,但是上述2类算法涉及多次迭代,且不同图像最优迭代次数不同,导致最优迭代次数的未知,不适于实时图像处理。文献[10]采用改进的非线性复扩散和自适应冲激滤波器,提出了一种自适应图像放大算法,该算法根据像素局部方差进行自适应非线性复扩散,耦合局部方差约束的冲激滤波器增强模糊的图像边缘和细节,对图像有较好的放大效果,但是计算量大,而且对于图像缩小及图像的任意倍数放大不适用。

本文提出了一种基于 Lanczos 核的自适应实时图像插值算法,该算法基于 Lanczos 核(kernel)生成应用于整个目标图像的核查找表,并且目标图像光栅扫描过程中,所有像素点的生成仅涉及定点小数运算,避免了浮点数的运算,提高了处理效率;同时,算法采取了优化的光栅扫描方法,内存利用效率高。

2 基于 Lanczos 核的缩放原理

考虑到计算复杂度和内存利用率,实时条件下对图像进行缩放处理,通常是对二维图像依次在横向(X方向)和纵向(Y方向)分别进行插值处理。由于X方向和Y方向的缩放因子是固定参数,为了避免不必要的重复计算,本文算法预先计算出目标图像所有像素的 Lanczos 核,并将其存储于核查找表中。当对目标图像进行光栅扫描时,直接从核查找表中读取 Lanczos 核,进行插值滤波,而不需要重新计算目标图像每一个像素点的 Lanczos 核。

2.1 总体算法

由于生成目标图像的每个像素颜色值都必须通过插值滤波生成,插值滤波必然是图像插值的性能瓶颈所在,为了提高图像插值效率,本文提出的算法中,插值滤波仅涉及定点小数运算,不涉及浮点数运算。该算法具体包括以下3个步骤,各步骤将在2.2~2.4节中进一步阐述。

1) 确定图像插值的横向和纵向缩放因子并将其转换成定点小数。

2) 基于 Lanczos 核生成横向和纵向各17个亚像素点对应的核查找表,并将核查找表中的 Lanczos 核转换成定点小数。

3) 按特定顺序完成目标图像的扫描及基于定点小数的9阶插值滤波,从而生成目标图像的像素颜色值。

2.2 缩放因子确定及定点化

为了避免在目标图像光栅扫描过程中使用浮

点运算,本文算法首先计算缩放因子并对其进行定点化处理。伪代码可以描述为 $S = \frac{(A-1) \ll 16}{B-1}$, 其

中, A 为源图像的宽度或高度, B 为目标图像的宽度或高度, S 为 X 方向或 Y 方向缩放因子; A 、 B 和 S 均为 32 位无符号整型变量,“ \ll ”为逻辑左移运算。处理后,缩放因子的高 16 位为整数部分,低 16 位为小数部分,即缩放因子为 I16F16 定点小数格式。由于缩放因子包含 16 位小数部分,其精度可以达到 2^{-16} , 即 $\frac{1}{65536}$, 可以满足任意倍数缩

放需求。

2.3 基于 Lanczos 核的目标像素核查找表生成

本节以横向核查找表的生成方法为例加以说明,纵向核查找表的生成方法与横向核查找表的生成方法类似。目标图像的某一像素由源图像的9个像素插值得出,如图1所示。三角形标示的是目标图像像素点对应的源图像中心像素点,左右4个小圆表示以源图像中心像素点为中心的左右相邻各4个像素点。

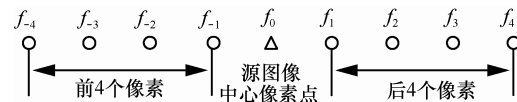


图1 9阶插值滤波示意

文献[11]指出 Lanczos 核^[12]是减少锯齿效应(aliasing)、尖锐效应(sharpness)和振铃效应(ringing)的最好折中(相对于其他窗 sinc 滤波器)。Lanczos 核的计算式为

$$\omega(d) = \begin{cases} 1, d = 1 \\ \frac{\sin(\pi d)}{\pi d} \cdot \frac{\sin\left(\frac{\pi d}{r}\right)}{\frac{\pi d}{r}}, 0 < |d| < r \\ 0, |d| \geq r \end{cases} \quad (1)$$

其中, d 表示图1所示9个像素点与亚像素点的距离, r 表示插值半径,由于是9阶插值滤波, r 取值为4。

每个亚像素点对应9个 Lanczos 核,本文亚像素点对应的坐标为 I16F16 定点小数格式,如果考虑所有亚像素点,则一共需要 65536×9 个 Lanczos 核,核查找表存储空间需要 1.125 MB。为减少核查找表存储空间及提高处理效率,本文算法仅截取亚像素

点的 5 位小数, 从而给出 32 个亚像素点的 Lanczos 核, 核查找表存储空间仅需要 576 B。32 个亚像素点如图 2 所示, 亚像素点的间距为 $\frac{1}{32}$, 亚像素从左至右依次为 0~0.968 75。由于目标图像像素原点坐标已有 0.5 的偏移 (见 2.4 节), 所以图 2 中三角形 (第 16 个亚像素, 从 0 开始计数) 示意的就是中心像素点。

对于第 i 个 (i 取值范围为 0~31) 亚像素点, 其对应的 9 个 Lanczos 核为

$$W(i) = \left[\omega\left(\left(\frac{16-i}{32}-4\right)S'\right) \dots \omega\left(\left(\frac{16-i}{32}\right)S'\right) \dots \omega\left(\left(\frac{16-i}{32}+4\right)S'\right) \right]$$

其中, $S' = \begin{cases} 1, & \frac{B}{A} > 1 \\ \frac{B}{A}, & \frac{B}{A} \leq 1 \end{cases}$, A 为源图像的宽度, B 为目标图像的宽度。进一步地, 横向核查找表可表示为

$$T_h = \begin{bmatrix} W(0) \\ W(1) \\ \vdots \\ W(16) \\ \vdots \\ W(31) \end{bmatrix} = \begin{bmatrix} \omega\left(\left(\frac{16}{32}-4\right)S'\right) & \dots & \omega\left(\left(\frac{16}{32}\right)S'\right) & \dots & \omega\left(\left(\frac{16}{32}+4\right)S'\right) \\ \omega\left(\left(\frac{15}{32}-4\right)S'\right) & \dots & \omega\left(\left(\frac{15}{32}\right)S'\right) & \dots & \omega\left(\left(\frac{15}{32}+4\right)S'\right) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \omega\left(\left(\frac{0}{32}-4\right)S'\right) & \dots & \omega\left(\left(\frac{0}{32}\right)S'\right) & \dots & \omega\left(\left(\frac{0}{32}+4\right)S'\right) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \omega\left(\left(\frac{-15}{32}-4\right)S'\right) & \dots & \omega\left(\left(\frac{-15}{32}\right)S'\right) & \dots & \omega\left(\left(\frac{-15}{32}+4\right)S'\right) \end{bmatrix}$$

由于 $\omega(d) = \omega(-d)$, $W(31)$ 可以由 $W(1)$ 推导出

来, $W(30)$ 可以由 $W(2)$ 推导出来, 依此类推, 核查找表生成只需计算第 0~16 个亚像素点的 Lanczos 核即可。将每个亚像素点的 Lanczos 核归一化, 并将归一化后的 Lanczos 核用 16 位无符号整型数表示, 其最高位为符号位, 次高位为整型位, 剩余 14 位为小数位, 存储空间进一步减少到 306 B。

2.4 插值滤波及扫描

本节以目标图像横向像素的插值滤波为例加以说明, 纵向像素的插值滤波与横向像素的插值滤波类似。目标图像第 n 个横向像素对应的源图像横向亚像素点 $f_s(n)$ 采用 $f_s(n) = \begin{cases} C_D, & n = 0 \\ f_s(n-1) + S, & n > 0 \end{cases}$ 计算, 其中, C_D 为目标图像像素横向坐标原点, S 为图像横向缩放因子。取源图像横向亚像素点 $f_s(n)$ 的整数部分为源图像横向中心像素点, 为使目标图像质量更平滑, 如果对应的源图像横向亚像素点坐标小数部分大于或等于 0.5, 选择下一个源像素为源像素横向中心像素点。同时, 为了避免在每一个目标像素点上进行此次判断, 本文将目标图像横向像素坐标原点 (定点小数格式, 用 32 位无符号整型数表示, 高 16 位为整数, 低 16 位为小数) 加上 0.5, 即取 $C_D = 0.5$, 则 $f_c = \lfloor f_s(n) \rfloor$, f_c 为目标图像第 n 个横向像素对应的源图像横向中心像素点, $\lfloor \cdot \rfloor$ 为向下取整运算符, 所以, 图 1 所示的 9 个横向像素依次为: $\{f_c - 4, \dots, f_c, \dots, f_c + 4\}$ 。

对于彩色 RGB 图像, 需分别对 R、G、B 3 个通道进行插值滤波。以 R 通道为例, $R^D = \sum_{i=-4}^4 R_i^S \omega'_i$, 其中, R^D 为目标图像像素的红色分量值, R_i^S 表示以源图像横向中心像素点为中心的 9 个源图像横向像素的红色分量值, ω'_i 对应 9 个源图像横向像素点的插值权重, 其计算方法为

$$\omega'_i = \begin{cases} \omega_i, & f_s - \lfloor f_s \rfloor \leq 0.5, \text{由 } index \text{ 索引 } T_h \text{ 得到} \\ \omega_j, & f_s - \lfloor f_s \rfloor > 0.5, \text{由 } (32 - index) \text{ 索引 } T_h \text{ 得到} \end{cases}$$

(2)

其中, T_h 为横向核查找表, 包含图 2 所示第 0~16 个亚像素点的 Lanczos 核; $index$ 取 $f_s - \lfloor f_s \rfloor$ 小数部

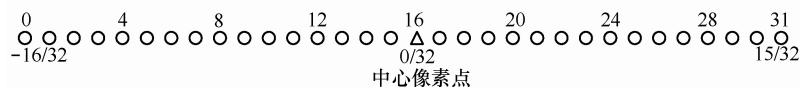


图 2 亚像素点示意

分的最高 5 位; ω_i 和 ω_j 表示某亚像素点对应的 9 个 Lanczos 核。 f_s 为某目标图像像素点对应的源图像横向亚像素点, f_s 小数部分小于或等于 0.5 时, 目标像素对应的源图像横向亚像素点的 9 个 Lanczos 核可以通过 $index$ 索引 T_h 获得, 当 f_s 小数部分大于 0.5 时, 目标像素对应的源图像横向亚像素点的 9 个 Lanczos 核则通过 $32-index$ 索引 T_h 并将得到的 9 个 Lanczos 核逆序取值获得。

源图像横向像素在内存中是按光栅扫描顺序 (由左至右, 由上至下) 存储的, 可以按照光栅扫描顺序依次读取源图像上的 RGB 值。纵向缩放如按正常的从上到下, 从左到右顺序扫描, 显然, 内存读写效率必定会成为处理效率的瓶颈。因此, 本文提出了一种高效的扫描方法, 如图 3 所示, 对于某一个固定的目标图像 Y 坐标, 本文算法一次扫描 16 个像素点 (假设一个缓存行为 64 B, 像素为 32 bit 色深), 称之为 span; 以 span 为单位, 从上到下扫描, 直至所有像素行扫描结束, 称之为 stripe; 以 stripe 为单位, 继续从左到右扫描, 直至所有像素列扫描结束。由于 span 中横向像素的地址连续性, 就可以保证内存读写效率的最大利用。另外, 纵向缩放时, 目标图像相邻像素点对应的部分源图像像素点同样可以被重复利用, 避免了内存的反复读写。

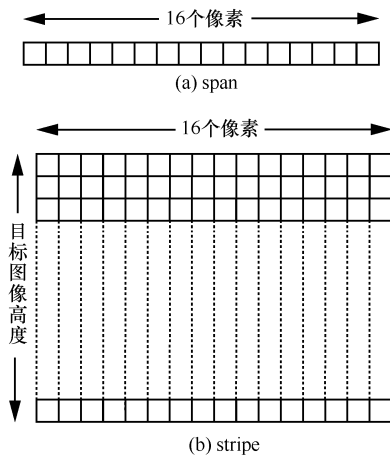


图 3 纵向光栅扫描

3 实验结果及分析

为了验证本文算法的有效性, 将本文算法与双线性插值、双三次插值及文献[4]算法进行比较。

3.1 客观质量评价

以标准图像测试图 Lena 彩色 RGB 图为准,

先对 Lena 原图进行 X 方向和 Y 方向的 $8^{\frac{1}{2}}$ 倍放大, 然后用同样算法将放大后的图像缩小到原图像尺寸 ($8^{\frac{1}{2}}$ 倍缩小)。Lena 彩色原图的彩色分辨率是 512×512 , 经放大后的临时图像分辨率为 1448×1448 , 再经缩小后的图像分辨率为 512×512 。峰值信噪比 (PSNR) 是衡量图像客观质量的主要指标, 其值越大, 表示目标图像越接近实际图像, 缩放质量也越高。表 1 对比了双线性插值、双三次插值、文献[4]算法和本文算法的 PSNR。对于彩色图像, 先独立分析 R、G、B 这 3 个通道的 PSNR, 再求平均值, 由实验数据可以看出, 本文算法效果好于双线性插值和文献[4]算法, 与双三次插值效果差不多。结果表明, 本文算法生成的图像质量较好。

表 1 PSNR 对比

算法	PSNR _R /dB	PSNR _G /dB	PSNR _B /dB	平均值/dB
双线性插值	42.361 42	43.292 04	42.365 66	42.673 04
双三次插值	47.974 21	47.979 19	47.968 03	47.973 81
文献[4]算法	43.664 90	43.536 90	43.832 30	43.678 03
本文算法	47.950 00	47.923 70	47.961 90	47.945 20

3.2 主观效果评价

图 4 所示为 4 种算法的主观视觉效果。图 4(a) 是 Lena 原始图像, 大小为 512×512 , 其中方框标示的区域大小为 128×128 , 此块区域细节比较丰富。图 4(b)~图 4(e) 分别为方框所示区域采用 4 种不同算法获取的放大图像, 大小均为 512×512 。双线性插值视觉效果和文献[4]比较平滑, 但稍显模糊; 双三次插值和本文算法的视觉效果较双线性插值平滑度稍差, 但显得比较清晰, 整体视觉效果层次分明。结果表明, 本文算法与双三次插值的主观视觉效果差异不明显。

3.3 处理性能评价

本文使用 C 语言实现了 4 种算法, 并在 Intel I7 2.67 GB, 4 GB 内存的机器上执行了 4 种算法程序, 对 Lena 512×512 彩色原图进行 X 方向和 Y 方向的 4 倍放大(图像放大更能体现算法的处理性能), 表 2 列出了 4 种算法程序的计算时间及为获取一个目标图像像素点所需的加法和乘法次数。由实验数据得出, 本文算法的计算时间仅为双线性插值的 33%, 其计算速度远远快于双三次插值, 而文献[4]算法效率较高, 但本文算法的处理效率仍然比文献[4]高。同时, 为了获取一个目标图像像素点, 本文算法的



图 4 主观视觉效果对比

加法和乘法次数大大少于双线性插值和双三次插值,也略少于文献[4]。另外,在 ALTERA STATIX II 实现本文算法,其时钟频率为 100 MHz,对 Lena 彩色原图 X 方向和 Y 方向的 4 倍放大,硬件仅消耗了 40 ms。

表 2 性能对比

性能评估	本文算法	双线性插值	双三次插值	文献[4]算法
计算时间/ms	172	515	671	261
加法次数	8	16	22	9
乘法次数	9	18	29	12

文献[4]提出的算法效率较高,表 3 列出了本文算法与文献[4]算法的进一步对比。本文算法中,缩小算法与放大算法一致。为了获取一个目标图像像素点,本文算法的加法和乘法次数都少于文献[4]算法;本文算法所需的插值权重的总数与原图像及目标图像的大小无关,而文献[4]的缩小算法所需的插值权重的总数为 $W_s \times 2 + H_s \times 2$ (其中, W_s 、 H_s 为原

表 3 本文算法与文献[4]算法的进一步对比

性能评估	本文算法	文献[4]的缩小算法	文献[4]的放大算法
加法次数	8	8	9
乘法次数	9	15	12
插值权重总数	306	$W_s \times 2 + H_s \times 2$	$W_d \times 2 + H_d \times 2$
像素索引总数	0	$W_s + H_s$	$W_d + H_d$
亚像素索引总数	0	$W_s + H_s$	0

图像的宽度和高度),文献[4]的放大算法所需的插值权重的总数为 $W_d \times 2 + H_d \times 2$ (其中, W_d 、 H_d 为目标图像的宽度和高度);本文算法不需要像素索引和亚像素索引,而文献[4]需要生成较多的像素索引。

4 结束语

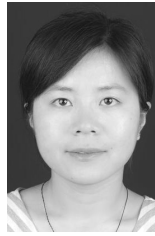
本文提出了一种实时彩色图像缩放算法,该算法基于 Lanczos 核生成应用于整个目标图像的核查找表,并且使目标图像光栅扫描过程中所有像素点的插值仅涉及定点小数运算,计算复杂度低;同时该算法采取了优化的光栅扫描方法,内存需求低,内存效率高。结果表明,该算法不仅获得了较好的图像质量,而且计算复杂度远低于双三次插值和双线性插值,便于硬件实现。综上所述,该算法能有效实现彩色图像缩放,特别适合于实时条件下的彩色图像缩放。

参考文献:

- [1] JAINA. Fundamentals of digital image processing[M]. Prentice-Hall, Inc Upper Saddle River, NJ, USA, 1989.
- [2] KEYS R. Cubic convolution interpolation for digital image processing[J]. IEEE Trans Acoust, Speech, Signal Processing, 1981, 29(6): 1153-1160.
- [3] XIAO J P, ZOU X C, LIU Z L. Adaptive interpolation algorithm for real-time image resizing[C]//Proceedings of the 1st International Conference on Innovative Computing, Information and Control. 2006.

- [4] PARKKINEN J, NENONEN P. A fast method for scaling color images[C]//17th European Signal Processing Conference, 2009: 2032-2036.
- [5] 刘晓松, 杨新, 汪进. 基于统计特征的彩色图像快速插值方法[J]. 电子学报, 2004, 32(1): 29-33.
LIU X S, YANG X, WANG J. Statistical feature based fast color image interpolation[J]. Acta Electronica Sinica, 2004, 32(1): 29-33.
- [6] 郎文辉, 江朝晖, 周荷琴, 等. 区域指导的彩色图像插值[J]. 电子学报, 2006, 34(1): 35-39.
LANG W H, JIANG Z H, ZHOU H Q, et al. Area-directed interpolation of color image[J]. Acta Electronica Sinica, 2006, 34(1):35-39.
- [7] FU S J, RUAN Q Q, WANG W Q. Region based shock-diffusion equation for adaptive image enhancement[C]//Advances in Machine Vision, Image Processing, and Pattern Analysis. Berlin: Springer, 2006: 387-395.
- [8] 付树军, 阮秋琦, 穆成坡, 等. 基于双向耦合扩散的保持特片的边缘锐化和图像增强[J]. 计算机学报, 2008, 31(3):529-536.
FU S J, RUAN Q Q, MU C P, et al. Feature preserving coupled bidirectional flow for edge sharpening and image enhancement[J]. Chinese Journal of Computers, 2008, 31(3):529-536.
- [9] 肖进胜, 饶天宇, 贾茜, 等. 改进的自适应冲击滤波图像超分辨率插值算法[J]. 计算机学报, 2015, 38(6): 1131-1139.
XIAO J S, RAO T Y, JIA Q, et al. Interpolation algorithm based on improved adaptive shock filter in image super-resolution[J]. Chinese Journal of Computers, 2015, 38(6): 1131-1139.
- [10] 席志红, 海涛. 基于非线性复扩散耦合冲激滤波器的图像放大算法研究[J]. 通信学报, 2014, 35(2): 1-8.
XI Z H, HAI T. Image enlargement research based on anisotropic complex diffusion coupling to shock filter[J]. Journal on Communications, 2014, 35(2): 1-8.
- [11] TURKOWSKI K, GABRIEL S. Filters for common resampling tasks[M]. Academic Press, 1990: 147-165.
- [12] CLAUDEE. Lanczos filtering in one and two dimensions[J]. Journal of Applied Meteorology, 1979, 18(8): 1016-1022.

作者简介:



郭莹(1986-), 女, 山东枣庄人, 武汉数字工程研究所工程师, 主要研究方向为数字图像处理、目标识别技术。

李伦(1986-), 男, 湖北武汉人, 武汉数字工程研究所高级工程师, 主要研究方向为图像处理、辅助决策。

王鹏(1980-), 男, 湖北枝江人, 武汉数字工程研究所高级工程师, 主要研究方向为图像处理、数据融合技术。