

一种图像快速线性插值的实现方案与分析

王 杰¹, 李洪兴², 王加银¹, 苗志宏³

(1. 北京师范大学数学科学学院, 数学与复杂系统教育部重点实验室, 北京 100875;

2. 大连理工大学电信学院, 辽宁大连 116023; 3. 中国人民武装警察部队学院消防工程系, 河北廊坊 065000)

摘 要: 几何变换是许多图像处理过程的基础. 由于线性插值需要较少的运算量, 又能在一定程度上保证图像质量, 因此, 图像几何变换常使用线性插值方法. 然而在需要实时处理的场合或运算能力较差的设备上, 线性插值的运算速度仍需提高. 本文首先总结了数字设备上运算整数化的两条规则. 根据这两条规则, 在不损失精度的条件下, 得到图像双线性插值的整数化方案(三线性插值可同样处理). 然后以几何变换中最常用的仿射变换为例, 给出整个图像变换过程的整数化方法及分析. PC 机上的实验证明, 本文方案相比原始方法执行性能大大提高. 此外, 本文的方案是高度可并行的.

关键词: 插值; 双线性; 三线性; 整数化

中图分类号: TN911.73 **文献标识码:** A **文章编号:** 0372-2112 (2009) 07-1481-206

A Scheme and Its Analysis for Fast Linear Image Interpolation

WANG Jie¹, LI Hongxing², WANG Jia2yin¹, MIAO Zh2hong³

(1. School of Mathematical Sciences, Beijing Normal University, Key Laboratory of Mathematics and Complex System, Ministry of Education,

Beijing 100875, China; 2. School of Electronic & Information Engineering, Dalian University of Technology, Dalian 116023, China;

3. Department of Fire Protection Engineering, The Chinese People's Armed Police Force Academy, Langfang, Hebei 065000, China)

Abstract: Geometry transformation is the basis of many image processing procedure. The linear interpolation method need little computation and can maintain image quality to some extent, so it is selected for the image geometry transformation in many cases. But the linear interpolation still need to be speed up when realtime processing is needed or being computed on devices with poor computing capacity. In this paper, two rules are concluded to integerize operands of an expression computed on digital devices. Based on the two rules, a scheme is proposed to integerize bilinear image interpolation (trilinear interpolation can be as the same), under conditions that the result of which has no loss in precision compared to the result of the original method. Then, the most frequently used transformation, affine transformation, is taken as an example to give a scheme and its analysis to integerize the whole image transformation procedure. The experiment on PC shows that the proposed scheme is much faster than the original method. In addition, the scheme proposed in this paper is highly parallel.

Key words: interpolation; bilinear; trilinear; integerize

1 引言

几何变换是许多图像处理过程的基础. 二维图像的双线性插值, 能得到比最近邻插值方法更好的效果, 而相比复杂插值方法, 如三次插值, 样条方法等, 只需更少的计算量^[1], 因此在图像处理过程中得到广泛应用^[2-5]. 对应的, 在三维空间中, 三线性插值可以被用于三维图像的变换.

数码相机等设备上的图像几何变换, 通常综合考虑运算能力和变换效果, 使用双线性插值^[4]. 在这类设备

上, 一般希望更快的速度和更低的功耗, 插值运算的性能仍需提高. 许多手持设备和数字电视芯片等也有类似的需求. 另外, 在有些需要实时处理的应用场合, 如医学超声实时宽景成像, 每秒需要处理十帧甚至几十帧图像, 线性插值往往成为速度瓶颈之一.

在提高双线性插值的速度上, 已有许多好的研究成果. 主要有两类方法: (1) 利用特殊硬件的特性, 如使用 PC 通用处理器的单指令多数据流 (SIMD) 特性, 对插值运算并行化^[6], 或使用显卡 GPU 的功能进行加速^[3]; (2) 针对一些应用的特殊条件, 给出特殊设计方案^[7,8].

本文不同于这两类方法,从通用角度出发,给出了一种整数化方案,在不改变原始插值精度的条件下,避免浮点运算,提高运算速度.在浮点运算能力较差的嵌入式设备上,更会有独到的优势.

整数化方案,是将参与运算的浮点数放大一定的倍数,然后舍去小数位,转化为整数.这样,接下来的插值运算过程都是整数运算,计算结果再缩小回正常范围,得到正确结果.这个思想可以用下面的简单运算加以说明:

$$0.251 @17 = 4.267, \text{取整结果为 } 4;$$

取放大倍数为 8, $\text{floor}(0.251 @8) @17 = 2 @17 = 34$, 再用 8 整除 34 也得到 4,

其中 $\text{floor}(X)$ 表示不超过 X 的最大整数,即下取整.整数化的运算结果要保证正确性,即运算数不能超出设备的表达能力,且相对原始运算不能损失精度.因此,需要确定合适的放大因子(如上面的 8),使得运算满足整数化的两条规则:

(1) 运算结果与理想结果相差(误差)小于给定值.如不大于 1 或 1/2,即相对先浮点运算然后取整的结果不损失精度.

(2) 运算过程中的数值不会超过数字设备中整数的表达范围,即不产生上溢出(运算都转化为整数,不再存在浮点的下溢出问题).

由于三线性插值与双线性插值的整数化及分析过程完全相同,本文只以双线性插值为例讨论.常用数字图像的像素值,是限定在小范围内的整数,一般取 0 至 255.彩色图像使用多个这样的值表示一个像素,每个值分别对应不同的颜色分量.由于各个颜色分量可以并行处理,步骤完全相同,因此这里只需要考虑一个分量的情况.图像双线性插值的结果值和原始像素值一样,也是限定在小范围内的整数.可以利用上面的思想,把运算过程整数化,而不损失运算精度.

2 双线性插值

数字图像的几何变换表达为

$$(Xc, Yc) = f(X, Y) \quad (1)$$

其中 (X, Y) 为原始图像中的像素坐标值, (Xc, Yc) 为变换到目标图像中的坐标值.例如,仿射变换是最常用的数字图像几何变换,从目标图像坐标计算原图像的坐标,表示为:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} \begin{pmatrix} xc \\ yc \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix} \quad (2)$$

二维数字图像进行几何变换的双线性插值过程如下:目标图像中的任一像素 $p(xc, yc)$, 其在原图像中对应的位置为 $(x, y) = f^{-1}(xc, yc)$. 数字图像的像素坐标 (xc, yc) 都是整数值,计算出的 (x, y) 是实数值,落在原

图内如图 1 所示的位置,其中:

$$(x_1, y_1) = \text{floor}(x, y)$$

floor 表示下取整,即恰好比 (x, y) 小的整数坐标值, $x_2 = x_1 + 1, y_2 = y_1 + 1$. 原始图像在图 1 中四个位置对应的像素值分别为

$$p(x_1, y_1) = p_1, p(x_2, y_1) = p_2,$$

$$p(x_1, y_2) = p_3, p(x_2, y_2) = p_4.$$

设 $a = x - x_1, b = y - y_1$, 有 $1 > a, b \setminus 0$, 则双线性插值得到

$$\begin{aligned} p(xc, yc) &= p(x, y) \\ &= [p_1(1-a) + p_2a](1-b) \\ &\quad + [p_3(1-a) + p_4a]b \end{aligned} \quad (3)$$

结果中要对 $p(xc, yc)$ 舍入取整.

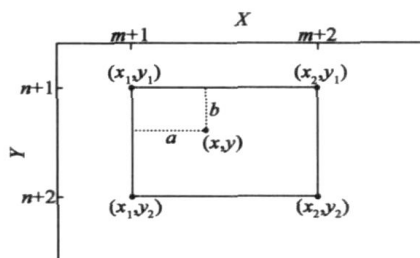


图 1 双线性插值的图示

3 整数化方案与误差分析

首先得出后面的分析过程中要用的一个结论,接着给出双线性插值的整数化运算过程,然后根据引言中的运算整数化的两条规则,推导出放大因子的合理取值.

3.1 整数化运算的截断误差

设 f 为一个参加运算的数,令

$$\underline{f} = \text{floor}(f @M) / M$$

其中 M 为正整数,是给定的放大因子, floor 表示下取整. \underline{f} 表示了 f 放大 M 倍后,转成整数的过程中,有效表达的截断结果.令

$$\$f = f - \underline{f}$$

表示这种截断的误差.

显然 $0 \leq f @M - \text{floor}(f @M) < 1$, 所以, $0 \leq f - \text{floor}(f @M) / M < 1/M$, 有

$$\underline{f} \leq f, 0 \leq \$f < 1/M \quad (4)$$

3.2 双线性插值运算的整数化

式(3)中的 a 和 b 都是浮点数,整数化第一步是先将其中的浮点数放大一定的倍数,转换成整数.式(3)中我们先把 a, b 放大 M 倍,并下取整为 A, B :

$$A = \text{floor}(a @M); B = \text{floor}(b @M)$$

其中 $\text{floor}(X)$ 表示不超过 X 的最大整数,即下取整.相应的 $1-a$ 对应为 $M-A$, 且 $1-b$ 对应为 $M-B$. 与式(3)对应有(式中的运算数都已经是整数):

$$p(\underline{xc}, \underline{yc}) = \{ [p_1(M - A) + p_2A](M - B) + [p_3(M - A) + p_4A]B \} / (M^2) \quad (5)$$

最后要对结果取整. 注意: $(M - A)$ 不同于 $\text{floor}((1 - a)M)$. 因为 $(1 - a) + a = 1$, 且 $(M - A) + A = M$, 但是 $A + \text{floor}((1 - a)M)$ 不一定等于 M . 运算中使用 $(M - A)$, 而不是 $\text{floor}((1 - a)M)$, 保证结果的归一化.

实际计算时, 为减少计算量, 式(5)形式改为: 约定

$$p_{xy1} = p_1M + (p_2 - p_1)A \text{ 且 } p_{xy2} = p_3M + (p_4 - p_3)A, \\ p(\underline{xc}, \underline{yc}) = [p_{xy1}M + (p_{xy2} - p_{xy1})B] / (M^2) \quad (6)$$

式中最后的除法可以直接使用整数的整除, 得到整数结果, M 取为 2 的幂, 可以用移位运算代替乘法, 提高运算速度. 类似式(5)到式(6)的变形, 式(3)也可用同样变形减少计算量. 这里使用式(3)和式(5)进行分析, 只是为了便于推导.

3.1.3 放大因子 M 的取值范围

需要找到合适的 M , 使整数化后的运算结果与理想的浮点结果相差不大于 $1/2$ 或 1, 本文按照误差限制为 $1/2$ 进行推导 (误差限制放松为 1 时过程完全相同, 可以得到更广的取值范围). 即找到 M 的范围, 使

$$|p(\underline{xc}, \underline{yc}) - p(\underline{xc}, \underline{yc})| \leq 1/2 \quad (7)$$

式(5)的整数化过程可以看作是对理想浮点数的截断, 令

$$\underline{a} = A/M = \text{floor}(a @ M) / M, \\ \underline{b} = B/M = \text{floor}(b @ M) / M$$

则式(5)变为

$$p(\underline{xc}, \underline{yc}) = [p_1(1 - \underline{a}) + p_2 \underline{a}](1 - \underline{b}) + [p_3(1 - \underline{a}) + p_4 \underline{a}] \underline{b} \quad (8)$$

令

$$\$a = a - \underline{a} = a - \text{floor}(a @ M) / M, \\ \$b = b - \underline{b} = b - \text{floor}(b @ M) / M \quad (9)$$

其中 $\$a, \b 表示了 $\underline{a}, \underline{b}$ 对 a, b 的运算截断误差. 有

$$p(\underline{xc}, \underline{yc}) = p(\underline{xc}, \underline{yc}) + [(p_1 - p_3)(1 - \underline{a}) + (p_2 - p_4) \underline{a}] \$b + [(p_1 - p_2)(1 - \underline{b}) + (p_3 - p_4) \underline{b}] \$a \\ \text{由式(4)的结论, } 0 \leq \$a, \$b < 1/M, \text{ 简单推导可得} \\ |p(\underline{xc}, \underline{yc}) - p(\underline{xc}, \underline{yc})| \leq [|\max\{p_1 - p_3, p_2 - p_4\}| \$b + |\max\{p_1 - p_2, p_3 - p_4\}| \$a] \quad (10)$$

综合式(7)和式(10), 只要取合适的 M , 使

$$[|\max\{p_1 - p_3, p_2 - p_4\}| \$b + |\max\{p_1 - p_2, p_3 - p_4\}| \$a] \leq 2 @ \max\{|\max\{p_1 - p_3, p_2 - p_4\}|, |\max\{p_1 - p_2, p_3 - p_4\}|\} @ \max\{\$a, \$b\} \leq 1/2 \quad (11)$$

成立. 这里以最为普遍使用的像素值范围 (0 至 255) 为例进行分析, 即 $p_i \in [0, 255]$, ($i = 1, 2, 3, 4$), 且为整数, 有

$$\max\left\{ |\max\{p_1 - p_3, p_2 - p_4\}|, |\max\{p_1 - p_2, p_3 - p_4\}| \right\} \leq 255,$$

(等号成立的情况存在, 如 $p_1 = 255$, 其它都为 0). 因此只要使

$$\max\{\$a, \$b\} \leq 1/1020 \quad (12)$$

成立. 由式(9), 式(4), 只要 $1/M \leq 1/1020$, 即取 $M \geq 1020$ 可使结果误差不大于 $1/2$.

下面推导使 M 符合整数化的第二条规则, 不产生上溢出的范围. 式(5)中

$$[p_1(M - A) + p_2A](M - B) + [p_3(M - A) + p_4A]B \leq \max\{p_1, p_2, p_3, p_4\} @ M^2 \leq 255 @ M^2$$

在 32 位设备上, 带符号整数上限为 $2^{31} - 1$, 因此, 只要 $M < \sqrt{(2^{31} - 1) / 255} \approx 2901.98$.

按照引言中的两条规则, 通过前两节的讨论可以得到, 在 32 位的设备上:

$$1020 \leq M \leq 2901$$

上式是误差不大于 $1/2$ 时的结果, 误差限为 1 时 1020 可对应改为 510, 在 64 位设备上的上限比 2901 更大. 实际运算中, 取

$$M = 2^{10} = 1024$$

(若误差限制为 1, 可取 512), 可以利用数字设备的位运算能力, 提高运算速度.

4 使用双线性插值的二维图像仿射变换的整数化方案

二维数字图像仿射变换的插值计算中, 虽然坐标变换式(2)的计算量小于插值式(3)的计算量, 但是计算次数相当. 因此如果把这个计算过程也整数化, 速度会有更大的提高. 本节首先给出仿射变换的整数化计算过程, 然后进行误差和有效范围的分析, 最后根据 32 位设备的特点, 特别给出一个可行的方案.

4.1.1 仿射变换整数化计算过程

在式(2)中 $\underline{xc}, \underline{yc}$ 表示数字图像的像素坐标值, 是整数. 类似于插值过程的整数化, 令放大因子为

$$N = M @ R \quad (13)$$

其中 M 对应上面第 3 节中插值计算使用的放大因子, R 为整数, 令

$$F_i = \text{floor}(f_i @ N), \quad i = 1, 2, 3, 4;$$

且

$$T_x = \text{floor}(t_x @ N); T_y = \text{floor}(t_y @ N)$$

整个图像的仿射变换过程可以表示为如下四个步骤:

(1) 仿射变换坐标计算:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} F_1 & F_2 \\ F_3 & F_4 \end{pmatrix} \begin{pmatrix} \underline{xc} \\ \underline{yc} \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \quad (14)$$

分别代入式(14)计算得到 X_H , X_L 和 Y_H , Y_L , 令 M 按照 313 节中取值, 若 $U = M @ Q$, Q 为整数, 则

$$\begin{cases} X = \text{floor}(X_H / Q) + \text{floor}(X_L / (VQ)) \\ Y = \text{floor}(Y_H / Q) + \text{floor}(Y_L / (VQ)) \end{cases} \quad (25)$$

若 $M = U @ Q$, Q 为整数, 则

$$\begin{cases} X = \text{floor}(X_H @ Q) + \text{floor}(X_L @ Q / V) \\ Y = \text{floor}(Y_H @ Q) + \text{floor}(Y_L @ Q / V) \end{cases} \quad (26)$$

在图像宽高较小时(如不超过 2^{14}), 可以取 $U = M$ 以减少运算量. 式(15)和式(16)中的变量可以如下计算

$$\begin{cases} x_i = \text{floor}(X / M) \\ A = X - x_i @ M \end{cases}$$

其他变量的计算过程相同. 最后使用式(6)(等价于式(5))进行插值计算. 这里 U , V 和 Q 都取为 2 的幂, 上面的运算过程可以用位运算加以改进, 保证快速的运算, 如 A 可以表达为 X 和 $M - 1$ 的按位与.

U , V 分别对应了运算的高部分 0 和低部分 0. 其中 U , V 的下限是保证 $U @ V$ 符合式(22)中 N 的范围. 上限是 U 和 V 分别保证符合式(23)中 N 的范围. 注意对 V , 式(23)中的 f_1, f_2, f_3, f_4 要换成 $f_i @ U - F_{iH}$, 同样的 t_x, t_y 要换成 $t_j @ U - T_{jH}$, 显然替换后的值都小于 1.

5 并行性的讨论

文献[6]利用彩色图像三个通道插值计算的并行性, 用 PC 的扩展指令(SIMD 指令)进行加速. 本文中的方案仍然可以依照这种并行方法. 其他部分使用 PC 的扩展指令加速有一定的困难, 但是在专用芯片上可以充分利用其并行性, 原始插值方法中所有可并行之处, 本文方案都同样适用. 本方案还具有额外的并行性, 如: 4.3 节中的 32 位设备上的方案, 高部分 0 和低部分 0 可以并行进行计算, 此部分的速度可以大约再提高一倍. 插值计算式(6)中的前两个式子的计算也可以并行进行.

6 实验结果

数字设备的浮点运算相对理想结果也是有误差的, 本文分析的是整数化运算相对于理想结果的误差. 因此, 实验中整数化插值的结果图像和浮点运算插值的结果图像的对应像素值最大相差可以为 2, 在边界处, 个别情况下变换结果图像的有效范围会相差一个像素. 对实验结果的正确性检查以此为依据.

实验分别在两台 PC 机上进行, 操作系统为 Windows XP. 机器 A 为 733MHz 的奔腾处理器, 系统运行后空闲内存约 30M. 机器 B 为 2.4GHz 的赛扬处理器, 系统运行后空闲内存约 302M. 算法使用 Visual C++ 6.0 实现. 算法实现中, 待比较方案的运算过程保持一致. 实

验方法是, 对一个较大的图像(宽 4500 像素, 高 3000 像素), 给定 3 组变换系数(变换后图的宽高不超过 16384), 分别使用原始方法和本文 4.3 节的方法进行仿射变换并计时. 用十次计时取平均值的办法得到运算速度. 分别检查两种方法的输出结果, 对应点像素值最大相差不超过 2, 只在很少的几个边界点上, 结果图像的有效范围偏差一个像素, 说明计算结果正确.

比较分析的运算速度如下表, 其中 /A 机原始 0 表示在机器 A 上的原始实现方法运行时间, /A 机整数 0 表示在机器 A 上本文方法运行时间. 相对原始实现, 本文方案的速度要快很多.

表 1 原始方案和本文方案的运行速度比较

	A 机原始	A 机整数	A 加速比	B 机原始	B 机整数	B 加速比
第一组系数	7.39s	1.55s	4.76	2.94s	1.15s	2.55
第二组系数	5.37s	1.00s	5.37	1.31s	0.35s	3.74
第三组系数	7.24s	1.52s	4.76	3.15s	1.28s	2.46

7 结论

图像的线性插值, 在需要速度和质量折中的场合得到广泛应用. 在这些应用中, 有时速度还需要进一步提升, 已有的改进方案主要针对特殊平台或使用降低精度的方法. 本文总结了整数化的两条规则, 通过误差和整数表达范围的分析, 得到图像双线性插值和三线插值的整数化方案. 然后以最为常用的仿射变换为例, 给出图像变换整个运算过程的整数化方案及误差分析, 并针对 32 位设备的字长限制, 提出改进方案. 其他类型的变换可以用类似的方法进行整数化, 误差和表达范围的分析也可以仿照本文. 本文给出的方案及分析主要针对 PC 平台, 在专用芯片上, 可以进一步优化, 以降低功耗, 提高速度. 还可以考虑与文献[7-8]的工作结合, 提高径向畸变校正的速度.

参考文献:

- [1] Thomas M L, Claudia G, Klaus S. Survey: Interpolation Methods in Medical Image Processing [J]. IEEE Transactions on Medical Imaging, 1999, 18(11): 1049- 1075.
- [2] Fabio C. Using Bilinear Models for View-invariant Action and Identity Recognition [A]. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition [C]. New York, USA, 2006, Vol. 2, 1701- 1708.
- [3] Francis K, Anil K. Fast image interpolation for motion estimation using graphics hardware [A]. In: IS&T/SPIE Electronic Imaging 2004: Real Time Imaging [C]. 2004, Vol. SPIE 5297, 184 - 194.
- [4] 李奇, 徐之海, 冯华君. 用于数字相机的 CCD 颜色插值算法研究 [J]. 激光与红外, 2002, 32(5): 300- 302.

- LI Qi, XU Zhizhai, FENG HuaJun. Study of DSC s CCD Color Interpolation Algorithm[J]. Laser & Infrared, 2002, 32(5): 300- 302. (in Chinese)
- [5] 张树华, 陈晓冬, 刘琳波等. 基于 FPGA 的双线性内插方法实现 MEE 的畸变实时校正[J]. 传感技术学报, 2004, 17(4): 594- 599.
ZHANG Shuhua, CHEN Xiaodong, LIU Linbo, etc. FPGA Based MEE Distortion Correction RealTime System Design Using Bilinear Interpolation[J]. Chinese Journal of Sensors and Actuators, 2004, 17(4): 594- 599. (in Chinese)
- [6] James A, Kumar B. Applications Tuning for Streaming SIMD Extensions[Z], Intel Technology Journal 2nd quarter, 1999.
- [7] Gribbon K T, Bailey D G. A novel approach to real-time barrel interpolation[A]. In: Proceedings of 2nd IEEE International Workshop on Electronic Design, Test and Applications (DELTA. 04) [C]. Perth, Australia, 2004. 126- 131.
- [8] Gribbon K T, Johnston C T, Bailey D G. A real-time FPGA implementation of a barrel distortion correction algorithm with bilinear interpolation[A]. In: Proceedings of Image and Vision Computing New Zealand (IVCNZ. 03) [C]. Massey University, Palmerston North, New Zealand, 2003. 408- 413.

作者简介:



王杰 男, 1976年9月出生于江苏省连云港市, 北京师范大学数学科学学院博士生, 主要研究领域为图像处理与模式识别。
E-mail: sohawangjie@126.com



王加银 (通讯作者) 男, 1974年7月出生于四川省德阳市, 北京师范大学数学科学学院副教授, 主要研究方向为计算机辅助建模与仿真、模糊推理与模糊控制。
E-mail: wjy@bnu.edu.cn



李洪兴 男, 1953年7月出生于天津市, 大连理工大学电信学院教授, 博士生导师, 北方工业大学自动化系客座教授, 主要研究方向为模糊系统、智能控制、知识表示与数据挖掘等。
E-mail: lihx@dlut.edu.cn



苗志宏 男, 1964年4月出生于云南省迪庆市, 中国人民武装警察部队学院教授, 主要研究方向为鲁棒控制、机器智能与模糊控制。
E-mail: miaozhh@21cn.com