

文章编号 1674-2915(2009)06-0538-05

一种用硬件实现的 Bayer 格式图像恢复算法

李渊渊, 孟 伟

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院 研究生院, 北京 100039)

摘要:针对传统的双线性 Bayer 彩色图像恢复算法恢复效果不理想,边界部分不明显,局部图像有团块等问题,提出了一种基于硬件实现的 Bayer 图像快速插值算法。采用梯度算法对 Bayer 格式图像绿色通道进行恢复,根据像素点所属的颜色组对蓝色通道进行恢复。实验结果表明,本文算法比双线性法有更好的峰值信噪比(PSNR)值,RGB 3个通道的 PSNR 值均比双线性法高 5 dB 以上,而且算法消耗时间比双线性法少,恢复的图像视觉效果更好。实验处理一幅 512×512 的全彩图像仅需要 9.3 ms,完全可以满足实时性的要求,因此,本文算法在对实时性要求高的场合有很好的应用前景。

关键词: Bayer 图像;插值算法;现场可编程门阵列

中图分类号: TP391.4 **文献标识码:** A

Real-time hardware Bayer color image restoration algorithm and its implementation

LI Yuan-yuan, MENG Wei

(1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun 130033, China;*2. *Graduate University of Chinese Academy of Sciences, Beijing 100039, China*)

Abstract: An improved fast Bayer interpolation algorithm based on the Field Programming Gate Array(FPGA) was presented to solve the problems of a traditional bilinear Bayer color image restoration on edge blurring and lumps in images. With the algorithm, a gradient algorithm was used to restore the green channel and the color groups corresponding to pixels to restore the blue channel for a image. Experimental results indicate that the algorithm can offer a better visual effect of the images, and the Signal to Noise Ratios(SNRs) of RGB channels obtained from the proposed algorithm are all higher 5 dB than those of obtained from the bilinear algorithm. Moreover, the time-consuming of the algorithm is less than that of the bilinear algorithm, and a 512×512 full color image can be processed in 9.3 ms. These results can meet the requirements of real time image processing and have higher application values.

Key words: Bayer image; interpolation algorithm; Field Programming Gate Array(FPGA)

1 引言

数码相机、数码摄像机、摄像头等设备都采用 CCD 或 CMOS 作为图像传感器,而为了减小体积、降低成本,通常在 CCD 和 CMOS 之前加一个颜色滤波阵列(Color Filter Array)。绝大多数的 CCD 和 CMOS 图像传感器采用 Bayer 滤波阵列,所得到的 Bayer 格式图像由 RGB 三基色组成。在 Bayer 格式中,每个像素只表达一种颜色,因此为了得到全彩色的图像,需要利用其周围像素点的强度相关性来估计出缺失的另外两种颜色,这种处理称为彩色插值(Color Interpolation 或者 Demosaicking)。实现彩色图像恢复时,通常需要在专门的硬件上进行,随着半导体技术的发展,FPGA 作为可编程逻辑芯片,其在 SOC(System on Chip)中发挥着越来越重要的作用。在 Bayer 图像恢复中,FPGA 的使用也越来越广泛,本文算法即采用 FPGA 来实现。

2 插值算法

彩色插值的算法有很多,一般来说,算法可以分为两类。第一类是单个颜色通道独立插值算法,规律是未知的绿色分量值仅由已知的绿色像素的强度进行估算,对红和蓝色通道亦是如此。彩色插值算法中最典型、应用最多的是双线性法^[1]。如图 1 所示,其中 R_{34} 表示图像第 3 行第 4 列是红色分量, G_{24} 表示第 2 行第 4 列是绿色分量, B_{23} 表示第 2 行第 3 列是蓝色分量。

G_{11}	R_{12}	G_{13}	R_{14}	G_{15}
B_{21}	G_{22}	B_{23}	G_{24}	B_{25}
G_{31}	R_{32}	G_{33}	R_{34}	G_{35}
B_{41}	G_{42}	B_{43}	G_{44}	B_{45}
G_{51}	R_{52}	G_{53}	R_{54}	G_{55}

图 1 Bayer 彩色滤光阵列
Fig. 1 Bayer color filter array

第二类是利用多通道的相关性进行插值,这一类自适应的算法常常结合了对图像细节的分析判断以及颜色通道之间的相关性,插值效果要明显优于前一类。但算法复杂,例如边缘检测法和文献[2~5]中提到的算法等。也有利用小波算法来实现的,然而在硬件上实现时,复杂算法的实现将消耗大量的硬件资源,而且复杂算法大多是串行执行,这样就很难做到实时处理。所以,复杂算法并不适合硬件实现。

2.1 双线性插值算法

Bayer 格式图像中,图像的一半像素分给作为亮度信号的 G 通道,另一半像素分给作为色度信号的 R 通道和 B 通道^[2]。由于 G 通道的像素占图像总像素的一半,所以认为对 R 、 B 通道插值时根据 G 通道信息来获得,这样图像的质量会更好。这就是说,如果 G 分量的插值方法选择的好,不仅会提高 G 分量的图像质量,而且对 R 、 B 分量的插值有很大的帮助。因此,在彩色插值中, G 分量的插值相当于一个基准。

双线性插值算法属于单通道独立插值算法,其原理^[6]如下:每一坐标点的分量值分别由它相邻的位置分量值取平均得到,每个通道相互独立。以图 1 为例,考虑坐标点(3,4),此处只有红色信息,为了得到全彩信息,需要恢复出蓝色和绿色分量,利用双线性法可以得到:

$$G_{34} = \frac{G_{24} + G_{33} + G_{35} + G_{44}}{4}, \quad (1)$$

$$B_{34} = \frac{B_{23} + B_{25} + B_{43} + B_{45}}{4}.$$

由于双线性法是单通道之间独立插值,并且始终取 3×3 滤波器的平均值,忽略了细节信息,以及图片三个颜色通道之间的相关性,所以双线性法往往不能得到满意的插值效果,容易在细线状态结构的边缘处产生锯齿形图案。

2.2 改进的插值算法

基于双线性法的以上不足,本文提出一种新型恢复算法,对双线性法的不足进行改进。其中对 G 分量的恢复用的是梯度算法^[7],即保证插值沿着边界进行,不会跨越边界。在恢复绿色分量时,先通过计算梯度来检测边界方向,从而选择合适的插值方向。通过这样的改进,使算法在保证低复杂度的情况下,可以较好地恢复彩色图像,恢复出来的图片比双线性法有更好的边缘平滑度和色彩对比度,非常适合在 FPGA 上并行处理。具

体过程如下:

(1) 对绿色分量的恢复

以图1为例,假设 α 为该采样点水平方向的梯度, β 为竖直方向的梯度,

$$\begin{aligned} \alpha &= |G_{35} - G_{33}| \\ \beta &= |G_{24} - G_{44}| \end{aligned} \quad (2)$$

如果 $\alpha < \beta$,则认为水平方向边界的概率比较大,插值可能在水平方向进行,而水平方向的邻域像素与被插值点的像素的相关性也会大于竖直方向的像素。因此,

$$G_{34} = \frac{G_{33} + G_{35}}{2}, \quad (3)$$

如果 $\alpha > \beta$,则认为在竖直方向边界的概率大,插值在竖直方向进行,

$$G_{34} = \frac{G_{24} + G_{44}}{2}, \quad (4)$$

$$B_{34} = \begin{cases} \frac{B_{25} + B_{23}}{2} (b_1 < b_2) \text{ I } (b_1 < b_3) \text{ I } (b_1 < b_4) \\ \frac{B_{25} + B_{45}}{2} (b_2 < b_1) \text{ I } (b_2 < b_3) \text{ I } (b_2 < b_4) \\ \frac{B_{45} + B_{43}}{2} (b_3 < b_2) \text{ I } (b_3 < b_2) \text{ I } (b_3 < b_4) \\ \frac{B_{43} + B_{23}}{2} (b_4 < b_1) \text{ I } (b_4 < b_3) \text{ I } (b_4 < b_2) \end{cases} \quad (7)$$

(3) 红色分量不变,使用原值

3 硬件实现

随着可编程逻辑器件的发展,用FPGA芯片为核心的处理单元具有速度快、效率高、低功耗等特点,而且在以FPGA为核心的控制单元中,用FPGA完成Bayer图像的恢复可节省DSP硬件资源,使整体结构更简单。

3.1 系统总体框架

本系统以FPGA为控制、运算核心。在实际应用中,FPGA的输入端将与数字图像采集卡相连,而输出端则与传输数据的信道相连。本文采用片内预置数据模拟输入端,输出端通过USB接口芯片与PC机相连。图2为FPGA实现框图。

原始图像数据由外部输入FPGA后,主要经过4个阶段:(1)进入输入缓冲RAM暂存;(2)由内部寄存器读出,参与运算模块的计算;(3)计算

如果 α 为 β ,则认为该像素点处在一个平滑区域内,插值在整个邻域内进行,

$$G_{34} = \frac{G_{33} + G_{35} + G_{24} + G_{44}}{4}. \quad (5)$$

(2) 对蓝色分量的恢复

基本思想是先判断这个颜色属于上下左右中的哪个颜色族,然后再计算它的值,具体步骤如下:

$$\begin{aligned} b_1 &= B_{25} - B_{23} \\ b_2 &= B_{25} - B_{45} \\ b_3 &= B_{45} - B_{43} \\ b_4 &= B_{43} - B_{23} \end{aligned} \quad (6)$$

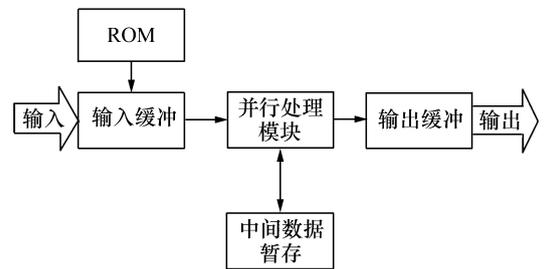


图2 系统框架图

Fig. 2 Chart of system framework

结果送入输出缓冲FIFO;(4)由输出时钟控制,输出到输出端口,从而完成整个系统的工作。

3.2 图像缓存与片内RAM使用

由于插值算法计算一个像素点需要用到邻近行充当像素值,而图像数据是连续输入的,所以对图像数据进行缓存是十分必要的。对于常见型号FPGA,假设图像数据每帧1280 bit × 1024 bit × 10 bit,缓存整帧数据是不现实的,也是不必要的。

本文算法是基于 3×3 掩模窗口进行计算,计算一个像素点最多用到相邻 3 行内的像素数据,故对实时图像无须存储整帧数据,只需缓存 3 行数据。输入缓冲由 3 个 16 kb 片内 RAM 组成,用于存储 3 行数据;输出缓冲用一个 4 kb 的 FIFO,总共使用 52 kb 片内 RAM,这样就无须使用外部存储器。

3.3 插值算法的 FPGA 实现

用 FPGA 实现插值算法,先将数据存入到 RAM 中,为提高速度向 14 个 RAM 并行输入数据,当向第 8 个 RAM 存入数据结束后,对数据进行处理并输出结果,同时向后 6 个 RAM 输入数据,数据输入后对 RAM7 ~ RAM14 中的数据进行处理并输出结果^[8],依次类推,用流水线的方式完成所有数据的处理(见图 3)。

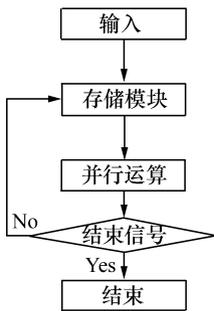


图 3 算法流程图

Fig. 3 Flow chart of algorithm

4 实验结果

4.1 实验结果比较

实验图片采用标准的全彩色图片。分别对图片进行双线性法恢复和本文算法恢复,结果如图 4 所示,从人眼视觉效果可以看出图片整体的插值效果好于双线性算法。用本文的插值算法图像亮度增加,对比度有明显提高,这使得边界更加突出。在平滑区域,双线性法恢复的图像可见明显的红格,而本文算法则不存在这个问题。

另外,实验还从图像的峰值信噪比(PSNR)方面对本文算法与双线性法的恢复图像进行了比较,实验结果如表 1 所示。

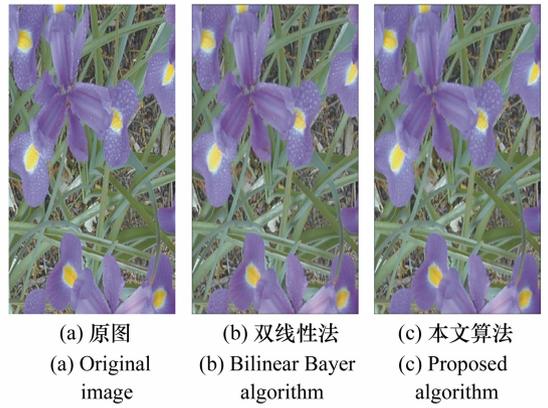


图 4 花朵图像对比图

Fig. 4 Comparison chart of flower image

表 1 PSNR 比较结果

Tab. 1 Comparison of PSNR

		花朵 (db)	人像 (db)
双线性	R 分量	26.06	25.33
	G 分量	26.44	26.09
	B 分量	23.48	25.70
本文算法	R 分量	30.86	28.35
	G 分量	31.00	30.12
	B 分量	29.54	29.65

从表 1 可以看出,与双线性法相比,本文算法 PSNR 值更高,进而说明了本文算法恢复出来的图像效果更好。



图 5 人像对比图

Fig. 5 Comparison chart of people image

4.2 算法实时性分析

在 FPGA 上进行本文算法计算时,需要同时

输入相邻上下两行的数据,计算的频率是数据输入的二倍。这样为了保证数据输入同时计算完所有输入数据,以本实验用的 CMOS MT9T001 为例,本文选取的花朵和人像两幅图片均为 512×512 大小全彩图像,分别如图 4、图 5 所示。分别用本文算法和双线性法进行恢复,试验结果如表 2 所示。

表 2 时间比较
Tab.2 Comparison of time

	花朵	人像
双线性法(ms)	10.6	10.2
本文算法(ms)	9.3	9.1

从表 2 可以看出 本文算法与双线性法相比,速度有一定的提高,处理速度达到 80 frame/s 以上,表明本算法在硬件上的处理速度完全可以满足实时性的要求。

5 结 论

根据硬件实现彩色图像恢复的实时性要求和双线性法恢复中存在的问题,本文提出了一种适用于硬件的彩色图像恢复算法,利用图像的梯度信息和相关区域的图像信息,进行彩色图像的恢复。实验结果表明,本文算法具有良好的性能。而且适用于硬件实时连续彩色图像恢复等对实时性要求高的场合。

参考文献:

- [1] GUNTURK B K, GLOTZBACH J, ALTUNBASAK Y, *et al.*. Demosaicking: color filter array interpolation[J]. *IEEE Signal Proc. Mag.*, 2005, 22(11):44-54.
- [2] 黄文增, 陈锦杏, 蔡政翰, 等. 高品质和快速彩色影像内插演算法之设计[J]. 台北科技大学学报, 2007, 40(2):13-25.
HUANG W T, CHEN C H, TSAI Z H, *et al.*. High-quality and high-speed demosaicking algorithm design[J]. *Taipei University Technol.*, 2007, 40(2):13-25. (in Chinese)
- [3] LUONG H, LEDDA A, PHILIPS W. *An Image Interpolation Scheme for Repetitive Structures*[M]. LNCS, Springer Berlin/Heidelberg, 2006, 441, 104-105.
- [4] VEGA M, MOLINA R, KATSAGGELOS A K. *Bayesian Reconstruction of Color Images Acquired with a Single CCD*[M]. LNCS, Springer Berlin/Heidelberg, 2005, 3522, 343-350.
- [5] COK D R. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal:US,4642678[P]. 1987-02-10.
- [6] HIBBARD R H. Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients:US, 5373322[P]. 1994-12-13.
- [7] LU W M, YAN Y P. Color filter array demosaicking: new method and performance measures[J]. *IEEE Trans. Image Proc.*, 2003, 12(10):1194-1210.
- [8] GUNTURK B K, ALTUNBASAK Y, MERSEREAU R. Color plane interpolation using alternating projections[J]. *IEEE Trans. Image Proc.*, 2002, 11(9):997-1013.

作者简介:李渊渊(1985—),男,河南南阳人,硕士研究生,主要从事机器视觉、数字图像处理方面的研究。

E-mail:tide713@163.com