

基于HLS的色彩插值算法硬件设计与实现

潘晓英, 李晨晨, 王昊, 薛玉锋

引用本文:

潘晓英, 李晨晨, 王昊, 等. 基于HLS的色彩插值算法硬件设计与实现[J]. 液晶与显示, 2020, 35(6): 595–603.

PAN Xiao-ying, LI Chen-chen, WANG Hao, et al. Hardware acceleration design and implementation of color interpolation algorithm based on HLS[J]. *Chinese Journal of Liquid Crystals and Displays*, 2020, 35(6): 595–603.

在线阅读 View online: <https://doi.org/10.3788/YJYXS20203506.0595>

您可能感兴趣的其他文章

Articles you may be interested in

基于Bayer格式图像的实时色彩还原算法

Real-time color reduction algorithm based on Bayer format

液晶与显示. 2019, 34(3): 310–316 <https://doi.org/10.3788/YJYXS20193403.0310>

低复杂度Bayer图像色彩重建算法

Low complexity demosaicing algorithm for Bayer pattern image

液晶与显示. 2016, 31(2): 179–184 <https://doi.org/10.3788/YJYXS20163102.0179>

实时视频流缩放系统设计

Design of real-time video stream scaling system

液晶与显示. 2019, 34(11): 1124–1130 <https://doi.org/10.3788/YJYXS20193411.1124>

应用先验插值校正 CT 金属伪影

Metal artifact reduction in computed tomography based on prior interpolation

液晶与显示. 2015(6): 1032–1039 <https://doi.org/10.3788/YJYXS20153006.1032>

基于Wu反走样的三角形光栅化边缘反走样算法

Edge anti-aliasing algorithm of triangle rasterization based on Wu algorithm

液晶与显示. 2015, 30(1): 163–169 <https://doi.org/10.3788/YJYXS20153001.0163>

文章编号:1007-2780(2020)06-0595-09

基于 HLS 的色彩插值算法硬件设计与实现

潘晓英¹, 李晨晨^{1*}, 王 昊², 薛玉锋¹

(1. 西安邮电大学 计算机学院, 陕西 西安 710121;

2. 厦门优莱柏网络科技有限公司, 福建 厦门 361008)

摘要:为了快速、高效地在 FPGA 平台上实现色彩插值算法并得到高质量的彩色图像,使用 Xilinx 高层次综合工具 HLS 设计并实现一种改进的色彩插值算法。该算法融合双线性插值算法和一阶微分边缘导向插值算法完成图像色彩插值,对以 R 分量与 B 分量为中心像素点的像素使用一阶微分边缘导向插值算法插值计算缺失的颜色分量,对以 G 分量为中心像素点的像素使用双线性插值算法插值计算缺失的颜色分量。该算法充分考虑图像边缘插值效果与 FPGA 硬件资源占用量,可对 Bayer 格式图像进行还原,恢复全彩色图像。本文使用 HLS 开发,与传统 FPGA 开发相比缩短了开发周期,提高了开发效率。在 Kodak 数据集上的测试结果表明,改进的色彩插值算法的彩色峰值信噪比(CPSNR)相较于常规算法高 4~6 dB,且减少了图像锯齿现象和边缘模糊等问题,本文算法占用 FPGA 资源较少,可实现实时的图像色彩插值。

关键词:色彩插值; Vivado HLS; FPGA; 硬件加速

中图分类号: TP391.41 **文献标识码:** A **doi:** 10.3788/YJYXS20203506.0595

Hardware acceleration design and implementation of color interpolation algorithm based on HLS

PAN Xiao-ying¹, LI Chen-chen^{1*}, WANG Hao², XUE Yu-feng¹

(1. School of Computer Science and Technology,

Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

2. ULAB Co., Ltd., Xiamen 361008, China)

Abstract: To quickly and efficiently implement a color interpolation algorithm on the FPGA platform and get a high-quality color image, an improved color interpolation algorithm is designed and implemented using Xilinx high-level synthesis tool HLS. The algorithm combines the bilinear interpolation algorithm and first-order differential edge oriented interpolation algorithm to complete image color interpolation. The first-order differential edge oriented interpolation algorithm is used to calculate the

收稿日期: 2019-11-10; **修订日期:** 2020-03-01.

基金项目: 国家自然科学基金(No.61772417); 陕西省教育厅专项科研计划项目(No.17JK0687); 西安市科技创新引导项目(No.201805040YD18CG24(7))

Supported by National Natural Science Foundation of China(No.61772417); Special Scientific Research Plan Project of Shaanxi Provincial Department of Education(No.17JK0687); Xi'an Science and Technology Innovation Guidance Project(No.201805040YD18CG24(7))

* 通信联系人, E-mail: lichen6435@163.com

missing color components for the pixels with R component and B component as the center pixel points, and the bilinear interpolation algorithm is used to calculate the missing color components for the pixels with G component as the center pixel points. The algorithm fully considers image edge interpolation and FPGA hardware resources, which can restore a Bayer format image and restore full-color image. Compared with the traditional FPGA development, this paper uses HLS to shorten the development cycle and improve development efficiency. The experimental results on the Kodak dataset show that the color peak signal-to-noise ratio (CPSNR) of the improved color interpolation algorithm is 4 ~ 6 dB higher than that of the conventional algorithm, and the image aliasing and edge blur are reduced. This algorithm takes up less FPGA resources and can realize real-time image color interpolation.

Key words: color interpolation; Vivado HLS; FPGA; hardware acceleration

1 引言

图像传感器是获取图像不可或缺的一个重要组成部分。目前市场上常见的图像传感器主要有互补金属氧化物半导体 (Complementary Metal Oxide Semiconductor, CMOS) 图像传感器和电荷耦合元件 (Charge-coupled Device, CCD) 图像传感器两种^[1]。由于制造工艺和原件成本等原因,市场上主流的传感器多为 CMOS 图像传感器。为获得彩色图像,通常需要在 CMOS 图像传感器表面覆盖一层彩色滤波阵列 (Color Filter Array, CFA),而 Bayer 格式的 CFA 是最常用的一种^[2],Bayer 格式的图像在每个像素位置上只有物理三基色 (红、绿、蓝) 中的一种色彩分量,要得到 RGB 彩色图像就需要根据该色彩分量周围的其他色彩分量插值计算出另外两种色彩分量,这种处理方式称为色彩插值^[3-4]。

随着图像传感器像素尺寸的增加,图像数据量也随之增大,使用软件方式实现色彩插值算法存在实时性差、效率低下等问题,现场可编程门阵列 (Field-Programmable Gate Array, FPGA)^[5-6] 在图像处理应用上具有处理数据量大、处理速度快等特点,使用 FPGA 可以加速实现色彩插值算法。色彩插值算法在实际硬件实现中多采用计算简单、实现较为容易的算法,比如邻近插值算法和双线性插值算法等^[7-9]。文献[10]提出在 FPGA 上采用邻近插值算法实现色彩还原,但图像色彩还原较差,细节丢失严重。文献[11]提出在 FPGA 上实现一种四合一图像插值算法,该算法 FPGA 实现简单,实时性较好,但图像边缘锯齿现

象较为严重。上述文献中插值算法的实现多使用硬件描述语言开发,对开发要求较高且开发时间较长。

针对上述算法存在图像色彩还原较差、细节丢失严重、存在锯齿现象以及开发周期较长等问题,本文提出了一种改进的色彩插值算法,该算法融合双线性插值法和一阶微分边缘导向插值法实现色彩插值,使用 Xilinx 高层次综合工具 HLS 完成算法开发。与传统 FPGA 开发使用硬件描述语言实现相比,HLS 使用 C/C++ 进行开发^[12-13],可降低开发难度,缩短开发时间。最后本文在 ZYNQ 平台上将 HLS 生成的色彩插值 IP 与图像采集模块结合实现了实时的图像色彩还原。

2 改进的色彩插值算法

本文的色彩插值算法采用双线性插值算法和一阶微分边缘导向插值算法相结合的方式实现图像色彩还原。如图 1 所示,在 Bayer 格式的 CFA 中,绿色分量占据所有像素信息的一半,所包含的图像信息量相对较多,需要在插值计算中更加精确地进行还原。当被插值像素点为 R 分量或 B 分量时,对缺失像素的插值使用较为复杂的算法计算。红色与蓝色分量分别占所有像素信息的 1/4,为减少算法复杂度与 FPGA 资源占用,当被插值像素点为 G 分量时,使用计算相对简单的算法完成插值计算。图 1 所示为 Bayer 格式的色彩滤波阵列。

在图像色彩还原时,如果插值算法没有考虑到图像边缘的特殊性,使用了跨边缘的图像像素

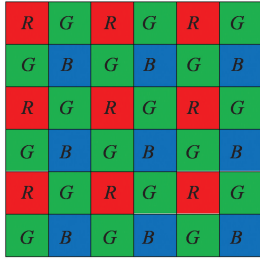


图 1 Bayer 色彩滤波阵列(CFA)

Fig.1 Bayer color filter array (CFA)

信息进行插值,就会导致插值后得到的图像边缘产生模糊和锯齿现象。改进后的色彩插值算法对以红色和蓝色分量为中心像素点的像素使用一阶微分边缘导向插值算法进行色彩插值还原,通过计算被插值像素点处水平(垂直)方向或对角方向的梯度值确定图像边缘,取梯度较小方向上的像素点作为估计点,从而避免了插值时错误地利用跨边缘的邻域像素信息,在插值计算时能有效还原图像边缘信息,改善图像边缘模糊以及锯齿现象等问题。对以绿色分量为中心像素点的像素使用双线性插值算法进行色彩插值还原,利用相邻像素中同色分量的像素值信息进行平均运算得到当前像素的插值色彩分量,该算法运算量较小,可有效减少 FPGA 资源的使用。

2.1 红色与蓝色分量处的色彩插值

如图 2(a)所示,当以蓝色分量 B_{22} 为中心像素点时,针对需要恢复的绿色分量和红色分量,本文使用一阶微分边缘导向插值算法进行色彩还原。对需要插值的绿色分量,首先计算水平方向和垂直方向上的梯度大小,取梯度较小方向上的像素点作为估计点,计算当前缺失像素, ΔH_g 表示水平梯度, ΔV_g 表示垂直梯度。

$$\begin{cases} \Delta H_g = |G_{21} - G_{23}| \\ \Delta V_g = |G_{12} - G_{32}| \end{cases}, \quad (1)$$

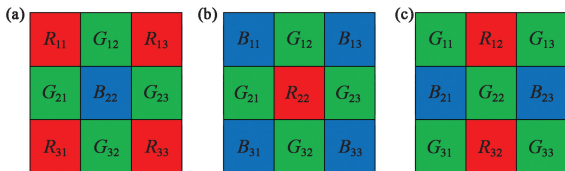


图 2 不同分量为中心像素点的示意图。(a)蓝色分量;(b)红色分量;(c)绿色分量

Fig.2 Diagram of different components as central pixels. (a) Blue component; (b) Red component; (c) Green component.

缺失的 G 分量插值计算方式如式(2)所示:

$$G_{22} = \begin{cases} \frac{G_{21} + G_{23}}{2}, & \Delta H_g < \Delta V_g \\ \frac{G_{12} + G_{32}}{2}, & \Delta H_g > \Delta V_g \\ \frac{G_{12} + G_{21} + G_{23} + G_{32}}{4}, & \Delta H_g = \Delta V_g \end{cases}, \quad (2)$$

对蓝色分量 B_{22} 进行插值求取红色分量 R , 根据蓝色分量 B_{22} 周围的红色分量的值来确定对角梯度值, H_r 表示主对角梯度, V_r 表示次对角梯度。取梯度较小方向上的像素点作为估计点, 对角梯度值定义如下:

$$\begin{cases} H_r = |R_{11} - R_{33}| \\ V_r = |R_{13} - R_{31}| \end{cases}, \quad (3)$$

R 分量的插值计算方式如式(4)所示:

$$R_{22} = \begin{cases} \frac{R_{11} + R_{23}}{2}, & H_r < V_r \\ \frac{R_{13} + R_{31}}{2}, & H_r > V_r \\ \frac{R_{11} + R_{13} + R_{31} + R_{33}}{4}, & H_r = V_r \end{cases}, \quad (4)$$

在图 2(b)中,红色分量 R_{22} 为中心像素,需要插值计算出绿色分量和蓝色分量,其计算过程与以蓝色分量 B_{22} 为中心像素点时的计算方式相同,首先计算 R_{22} 处水平与垂直梯度值,插值计算出缺失的 G 分量,再计算 R_{22} 处的对角梯度值,插值计算出缺失的 B 分量。

2.2 绿色分量处的色彩插值

如图 2(c)所示,当以绿色分量 G_{22} 为中心像素点时,使用双线性插值算法进行色彩还原,首先对中心像素点邻域内相同颜色分量求均值,然后将该值作为中心像素点缺失颜色分量值。

对图 2(c)中像素点 G_{22} 进行插值,对应的红色分量 R 和蓝色分量 B 的恢复公式如下:

$$R_{22} = \frac{R_{12} + R_{32}}{2}, \quad (5)$$

$$B_{22} = \frac{B_{12} + B_{23}}{2}, \quad (6)$$

本文提出了一种改进后的色彩插值算法实现图像色彩还原,对不同的颜色分量,使用相对应的色彩插值算法,利用图像边缘的梯度信息区分图像的边缘,恢复出的图像边缘细节较好,减少了锯齿与

模糊现象。

3 色彩插值算法的 HLS 实现

Vivado HLS 是 Xilinx 公司推出的高层次综合工具。与使用硬件描述语言开发 FPGA 不同, HLS 使用 C/C++ 进行开发, 通过高层次综合将其转换成 HDL 代码, 并将程序打包成 IP 核, 因此具有很强的灵活性, 可大幅度缩短开发周期, 同时也具有良好的加速功能。

3.1 使用 HLS 设计色彩插值 IP 核

使用 Vivado HLS 完成 Bayer 格式图像色彩还原的流程如图 3 所示。

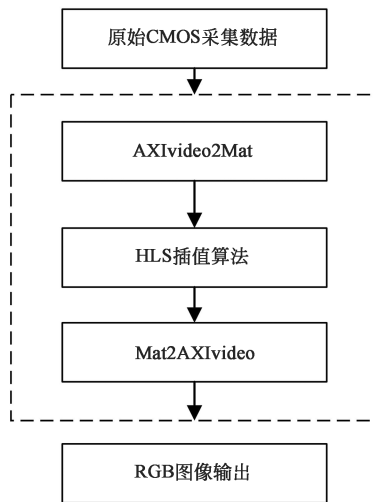


图 3 HLS 硬件加速流程

Fig.3 HLS hardware acceleration process

本文使用 Sony IMX 系列 CMOS 传感器完成图像采集, 图像尺寸为 2048×1536 。系统运行时, 摄像头完成取图, 原始 Bayer 格式图像数据流进色彩插值 IP, 通过 AXI4-Stream 协议将数据流转换为 HLS opencv 库下的 `hls::Mat` 格式, 然后进行色彩插值运算, 最后将处理完成的图像再经过格式转换流出色彩插值 IP, 完成 Bayer 格式图像的色彩插值, 最终得到 RGB 彩色图像。

使用 HLS 完成色彩插值算法的主要流程如图 4 所示。首先对图像进行格式转换, 根据 AXI4-Stream 协议将原始图像数据流转换为 HLS 视频库下的 `hls::Mat` 格式, 使用 `hls::LineBuffer` 缓存大小为 3×2048 的行缓存, 将行缓存中的数据使用 `hls::Window` 再缓存到一个

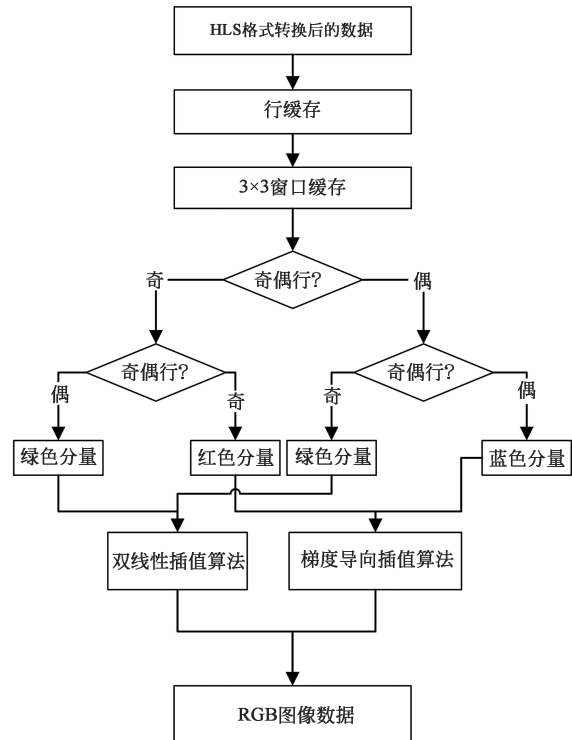


图 4 Vivado HLS 实现色彩插值算法流程框图

Fig.4 Vivado HLS algorithm flow chart

大小为 3×3 的滑动窗口, 对该 3×3 的滑动窗的中心像素点进行插值计算, 计算完成后窗口右移, 当该窗口完成行遍历后, 行缓存下移, 窗口继续向右移动, 直至完成所有像素点遍历。

如图 1 所示, 在 Bayer 格式的 CFA 中, 不同色彩分量有规律地分布, 其中 G 分量处于奇行偶列与偶行奇列, R 分量与 B 分量处于奇行奇列与偶行偶列。在色彩插值计算中, 需要判断该滑动窗中心像素点所处行列的奇偶性, 当中心像素处于奇行偶列与偶行奇列时, 对该像素采用双线性插值算法进行色彩插值。当中心像素处于奇行奇列与偶行偶列时, 对该像素采用一阶微分边缘导向插值算法进行色彩插值。

3.2 色彩插值 IP 硬件系统实现

在 Xilinx 高层次综合工具中, 将设计仿真通过的色彩插值算法打包成 IP 核, 在 Vivado 开发环境中将色彩插值 IP 添加进整体的图像采集系统完成色彩还原的硬件加速。图 5 为图像采集和色彩还原系统的整体设计。在整体系统设计中将摄像头采集到的 Bayer 格式数据输送到色彩插值 IP 中, 完成实时的图像色彩还原。

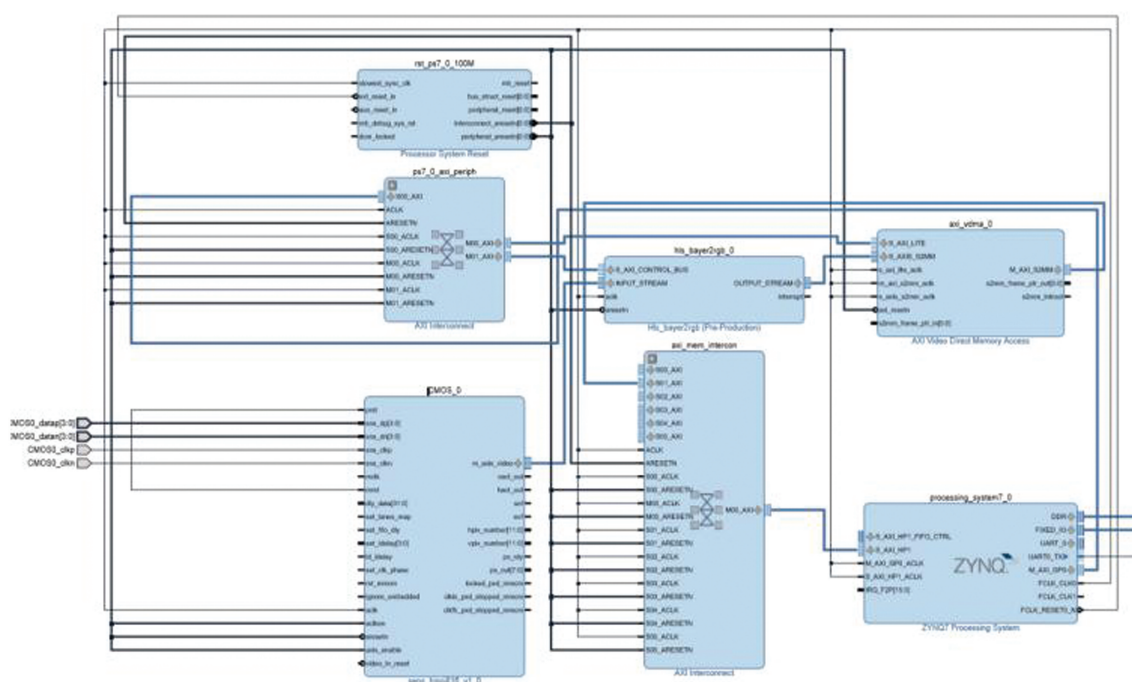


图 5 Vivado 中图像采集和色彩还原系统设计

Fig.5 Image acquisition and color restoration system

4 实验与结果分析

在对 Bayer 格式图像进行插值运算过程中有可能会出现伪彩色、摩尔纹现象、锯齿现象等问题。为衡量色彩插值算法的还原效果,需要对还原后的图像进行图像质量进行评价。衡量图像还原质量可分为主观评价和客观评价,主观判断不具有量化标准,因此本文选用了客观评价中评价

重构图像质量最常用的方法:彩色信噪峰值比(CPSNR)。CPSNR 值越大,说明插值后得到的图像质量越好。

式(7)所示为 CPSNR 的计算公式。其中: W 表示图片的宽度, H 表示图片的高度, $O_{i,j,k}$ 表示真实图片的原始值, $N_{i,j,k}$ 表示经过插值算法恢复后的图像, P 表示像素的峰值,因为本文图像格式为 8 bits,因此 P 取值为 255。

$$K_{\text{CPSNR}} = 10 \lg \left(\frac{P^2}{\frac{1}{3WH} \sum_{k=1}^3 \sum_{i=1}^W \sum_{j=1}^H (O_{i,j,k} - N_{i,j,k})^2} \right), \quad (7)$$

为验证本文算法的图像还原效果,随机选取经典的 Kodak 图像库里的 8 幅图像数据作为测试图

像,对选取图像进行图像还原处理,得到 Bayer 格式的图像再使用色彩插值算法进行还原。



(a)原始Kodak测试图像
(a)Original Kodak test image



(b)处理后的Bayer图像
(b) Bayer image after processing



(c)双线性插值还原图像
(c) Bilinear interpolation image

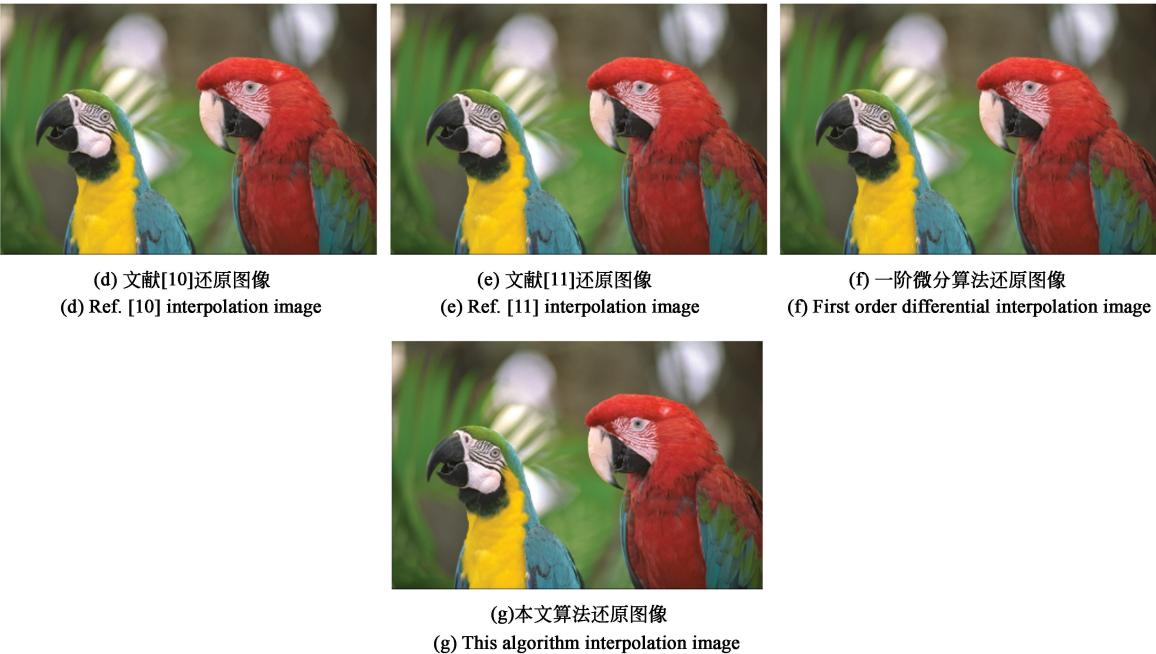


图 6 实验结果
Fig.6 Experimental results

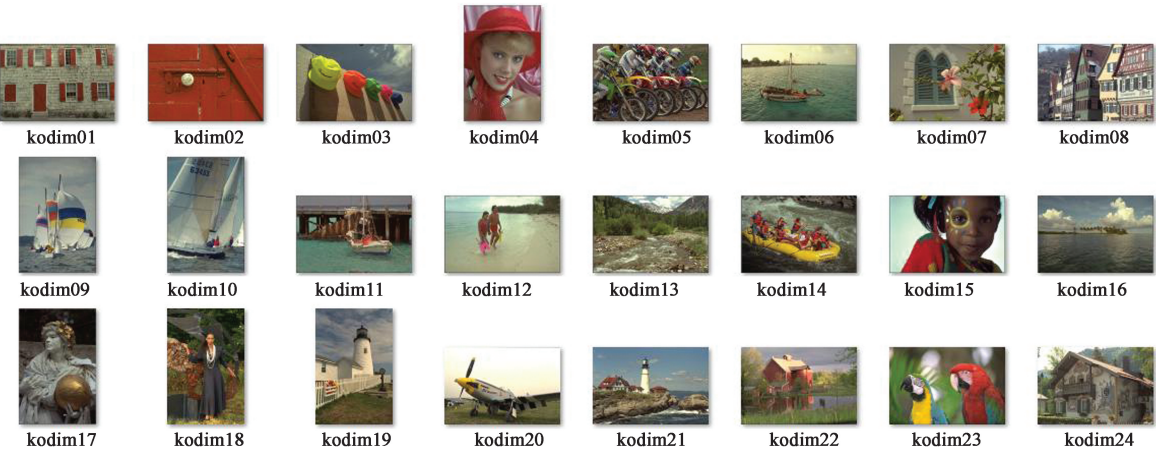


图 7 Kodak 测试图像集
Fig.7 Kodak image dataset

从图 6 中可以看出,采用本文算法插值还原得到的图像细节相较于双线性法图像细节更加丰富,色彩还原更好。相较于文献[10]与文献[11],图像边缘细节还原较好,锯齿现象较少。

表 1 几种算法的 CPSNR 值对比
Tab.1 Comparison of CPSNR values of three algorithms

	双线性算法	文献[10]	文献[11]	一阶微分算法	本文算法
图像 1	28.729 6	29.145 2	30.608 6	34.363 7	33.715 8
图像 2	31.156 1	30.563 9	33.485 2	35.376 4	34.869 8

续 表

	双线性算法	文献[10]	文献[11]	一阶微分算法	本文算法
图像 3	28.524 7	30.025 6	29.259 6	33.624 2	34.356 2
图像 4	25.142 9	29.120 3	30.452 2	35.401 2	34.758 1
图像 5	30.288 4	28.347 4	28.024 5	36.062 1	35.541 8
图像 6	27.346 7	29.982 5	30.453 3	34.856 3	33.658 2
图像 7	26.021 3	30.197 5	29.785 2	35.486 4	34.825 2
图像 8	29.789 4	30.424 6	31.668 7	37.168 7	36.487 1
平均值	28.374 89	29.725 88	30.467 16	35.292 38	34.776 53

表 2 硬件资源使用表

Tab.2 Hardware resource usage table

算法	BRAM_18K	FF	LTU
双线性插值算法	2	702	751
文献[8]	2	720	782
文献[9]	2	721	872
一阶微分边缘导向算法	4	1 034	1 626
本文算法	2	732	928

从表 1 中可以看出,本文算法的彩色峰值信噪比(CPSNR)相较于双线性插值算法、文献[8]、文献[9]所提算法平均高 4~6 dB,使用一阶微分边缘导向插值算法得到的 CPSNR 值比本文算法平均高 0.5 dB 左右。从表 2 中可看出,本文算法的 FPGA 资源使用略高于双线性法与文献[10]、文献[11]所提算法,其 FPGA 资源使用比 3 种算法平均高 13.7%,相较于一阶微分算法有大幅度下降,仅为一阶微分算法的 57.1%。

结合图像还原结果、CPSNR 数据以及 FPGA 的资源使用,改进后的色彩插值算法得到的图像质量

比常规插值算法还原出的图像质量要好。相较于一阶微分边缘导向插值算法,其使用 FPGA 资源使用大幅度减少。在保证图像还原质量的前提下,本文算法还原出的图像更加清晰,边缘细节较好,图像色彩还原真实,细节丰富,FPGA 资源使用较少。

最后,本文将设计验证完成的色彩插值 IP 进行板级验证,在 Zynq 平台上结合摄像头采集模组完成实时的图像色彩插值计算,如图 8 所示,图 8(a)为 Zynq 图像采集平台,图 8(b)为实际采集到的 Bayer 图像,图 8(c)使用本文色彩插值 IP 在 Zynq 平台上实际拍摄彩色还原图像。

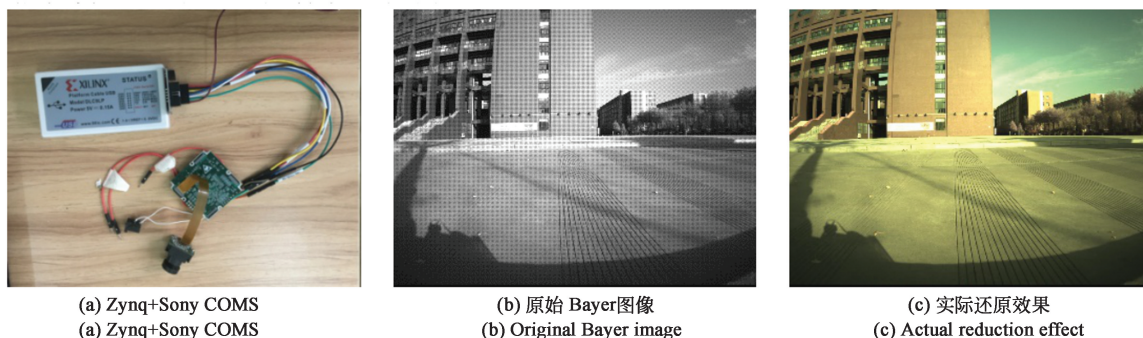


图 8 采集模块与板上还原结果

Fig.8 Acquisition module and restore results

5 结 论

提出了一种改进的色彩插值算法,该算法融合了双线性插值算法和一阶微分边缘导向插值算法。由于绿色分量所包含图像信息量较多,当被插值像素点为 R 分量或 B 分量时,使用一阶微分算法插值恢复缺失的颜色分量,可有效减少图像

边缘模糊与锯齿现象。考虑到 FPGA 资源使用与硬件成本,当被插值像素点为 G 分量时,使用双线性插值算法插值恢复缺失的颜色分量。实验结果表明,本文算法的彩色峰值信噪比相较于常规算法高 4~6 dB。使用 HLS 开发可快速完成算法的设计与实现,提高开发效率。将设计实现的算法打包生成 IP 核,可多次重复使用,提高了算法的复用性。

参 考 文 献:

- [1] 王世和,陈远金,刘彬.CCD 与 CMOS 国内外技术发展综述 [J].内燃机与配件,2017(13):112-114.
WANG S H, CHEN J Y, LIU B. CCD and COMS China and abroad overview of technology development [J]. *Internal Combustion Engine & Parts*, 2017(13): 112-114. (in Chinese)
- [2] HUANG Y S, CHENG S Y. Effective color-difference-based interpolation algorithm for CFA image demosaicking [C]//*Advances in Multimedia Modeling - 17th International Multimedia Modeling Conference*. Taipei, Taiwan, China: Springer, 2011: 457-469.
- [3] 王麓,徐弘基,王若溪,等.基于 Bayer 格式图像的实时色彩还原算法 [J].液晶与显示,2019,34(3):310-316.
WANG L, XU H J, WANG R X, *et al.* Real-time color reduction algorithm based on Bayer format [J]. *Chinese Journal of Liquid Crystals and Displays*, 2019, 34(3): 310-316. (in Chinese)
- [4] WANG D Y, YU G, ZHOU X, *et al.* Image demosaicking for Bayer-patterned CFA images using improved linear interpolation [C]//*2017 Seventh International Conference on Information Science and Technology*. Da Nang, Vietnam: IEEE, 2017: 464-469.
- [5] 贾建禄,赵金宇,王建立,等.基于 FPGA 的自适应光学波前处理算法 [J].光学精密工程,2017,25(10):2580-2583.
JIA J L, ZHAO J Y, WANG J L, *et al.* Adaptive optical wave-front processing algorithm based on FPGA [J]. *Optics and Precision Engineering*, 2017, 25(10): 2580-2583. (in Chinese)
- [6] 王向军,张继龙,阴雷.光流法运动估计在 FPGA 上的实现与性能分析 [J].光学精密工程,2019,27(1):211-220.
WANG X J, ZHANG J L, YIN L. Implementation and performance analysis of optical flow based motion estimation on FPGA [J]. *Optics and Precision Engineering*, 2019, 27(1): 211-220. (in Chinese)
- [7] 龚昌来,罗聪,杨冬涛.一种基于边缘方向的双线性插值方法 [J].激光与红外,2010,40(7):788-791.
GONG C L, LUO C, YANG D T. A novel bilinear interpolation method based on edge direction [J]. *Laser & Infrared*, 2010, 40(7): 788-791. (in Chinese)
- [8] BAILEY D, RANDHAWA S, LI J S J. Advanced Bayer demosaicing on FPGAs [C].2015 International Conference on Field Programmable Technology (FPT). Queenstown, New Zealand: IEEE, 2015: 216-220.
- [9] RANI K S, HANS W J. FPGA implementation of bilinear interpolation algorithm for CFA demosaicing [C].2013 International Conference on Communication and Signal Processing. Melmaruvathur, India: IEEE, 2013: 857-863.
- [10] 王健,应骏,曾维军,等.基于邻近插值法还原 Bayer RGB 的 FPGA 实现 [J].上海师范大学学报:自然科学版,2016,45(4):411-416.
WANG J, YING J, ZENG W J, *et al.* Implementation of restoring Bayer RGB on FPGA through nearest neighbor interpolation [J]. *Journal of Shanghai Normal University: Natural Sciences*, 2016, 45(4): 411-416. (in Chinese)
- [11] 刘亚亚.基于 FPGA 硬件的彩色图像恢复系统设计 [J].成都大学学报:自然科学版,2016,35(2):159-160.
LIU Y Y. Color image restoration design based on FPGA [J]. *Journal of Chengdu University: Natural Science Edition*, 2016, 35(2): 159-160. (in Chinese)
- [12] 党宏社,王黎,王晓倩.基于 Vivado HLS 的 FPGA 开发与应用研究[J].陕西科技大学学报:自然科学版,2015(1):

155-159.

DANG H S, WANG L, WANG X Q. Development and application of FPGA based onVivado HLS [J]. *Journal of Shaanxi University of Science & Technology: Natural Science Edition*, 2015(1): 155-159. (in Chinese)

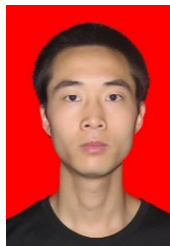
[13] 彭习武,张涛.基于 Vivado HLS 的边缘检测硬件加速应用 [J].电子技术应用,2017,43(5):70-73.

PENG X W, ZHANG T. Edge detection hardware acceleration based onVivado HLS [J]. *Application of Electronic Technique*, 2017, 43(5): 70-73. (in Chinese)

作者简介:



潘晓英(1981—),女,浙江丽水人,教授,硕士生导师,2009 年于西安电子科技大学获得博士学位,主要研究方向为人工智能、视觉信息处理等。E-mail: panxiaoying@xupt.edu.cn



李晨晨(1995—),男,陕西礼泉人,硕士研究生,2017 年于西安科技大学获得学士学位,主要研究方向为嵌入式图像处理与模式识别。E-mail: lichen6435@163.com