

## 基于FPGA的实时Bayer解马赛克算法与实现

张泽宇 张弘 伍凌帆 杨一帆 李旭亮

### FPGA-based real-time Bayer demosaicking algorithm and its implementation

ZHANG Zeyu, ZHANG Hong, WU Lingfan, YANG Yifan, LI Xuliang

引用本文:

张泽宇, 张弘, 伍凌帆, 等. 基于FPGA的实时Bayer解马赛克算法与实现[J]. 应用光学, 2022, 43(2): 240–247. DOI: 10.5768/JAO202243.0202002

ZHANG Zeyu, ZHANG Hong, WU Lingfan, et al. FPGA-based real-time Bayer demosaicking algorithm and its implementation[J]. *Journal of Applied Optics*, 2022, 43(2): 240–247. DOI: 10.5768/JAO202243.0202002

在线阅读 View online: <https://doi.org/10.5768/JAO202243.0202002>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### [基于FPGA的实时大气湍流图像复原算法及实现](#)

FPGA-based real-time atmospheric turbulence image restoration algorithm and its implementation

应用光学. 2021, 42(6): 1017–1024 <https://doi.org/10.5768/JAO202142.0602002>

#### [基于FPGA的雾霾视频图像实时复原系统研究](#)

Research and implementation of real-time fog and haze video image restoration system based on FPGA

应用光学. 2019, 40(5): 812–817 <https://doi.org/10.5768/JAO201940.0502004>

#### [基于FPGA的图像畸变矫正算法研究](#)

Image distortion correction algorithm based on FPGA

应用光学. 2020, 41(1): 86–93 <https://doi.org/10.5768/JAO202041.0102004>

#### [基于FPGA的自适应光学系统SPGD算法实现](#)

FPGA –based SPGD algorithm implementation of adaptive optical system

应用光学. 2021, 42(5): 810–816 <https://doi.org/10.5768/JAO202142.0501008>

#### [阵列细丝直径实时快速检测系统设计](#)

Real-time and fast detection system of arrayed filament diameter

应用光学. 2021, 42(6): 1011–1016 <https://doi.org/10.5768/JAO202142.0602001>

#### [基于离焦图像复原的光学设备实时视频对焦方法](#)

Real-time video focusing method for optical equipment based on defocus image restoration

应用光学. 2020, 41(1): 43–54 <https://doi.org/10.5768/JAO202041.0101007>



关注微信公众号，获得更多资讯信息

文章编号:1002-2082(2022)02-0240-08

# 基于 FPGA 的实时 Bayer 解马赛克算法与实现

张泽宇, 张 弘, 伍凌帆, 杨一帆, 李旭亮

(北京航空航天大学, 北京 102206)

**摘要:** Bayer 阵列被广泛地应用于 CMOS/CCD (complementary metal oxide semiconductor/charge-coupled device) 等前端传感器中, 用以对彩色图像进行压缩编码。通过解马赛克算法将 Bayer 阵列还原为红、绿、蓝彩色阵列, 算法性能影响着成像有效分辨率和纹理细节。随着半导体工艺的发展和目标识别等新型应用需求的提出, 图像设备向着高分辨率、低延迟的方向发展, 原有的解马赛克算法遇到性能瓶颈。提出了一种基于 FPGA (field programmable gate array) 的实时解马赛克算法, 能够准确地提取图像局部梯度方向并以引导实现色彩的插值复原, 整体算法仅需 7 行数据延迟。算法设计充分考虑 FPGA 的硬件特点, 设计了行缓存、梯度算子、梯度方向插值等模块, 降低硬件开销。实验表明, 本文算法在实现微米级延迟的同时, 保持了对图像纹理细节区域的复原效果, 在柯达数据集上的平均峰值信噪比达到 38.26 dB。

**关键词:** Bayer 阵列; 解马赛克算法; FPGA; 实时图像处理

中图分类号: TN391

文献标志码: A

DOI: 10.5768/JAO202243.0202002

## FPGA-based real-time Bayer demosaicking algorithm and its implementation

ZHANG Zeyu, ZHANG Hong, WU Lingfan, YANG Yifan, LI Xuliang

(Beihang University, Beijing 102206, China)

**Abstract:** The Bayer array is widely applied in the front-end sensors such as complementary metal oxide semiconductor/charge-coupled device (CMOS/CCD), in order to compress and encode the color images. The Bayer array is restored to red, green and blue color arrays by demosaicking algorithm, and the performance of the algorithm affects the imaging effective resolution and texture details. With the development of semiconductor technology and proposal of demand of new applications such as target recognition, the image devices move toward high resolution and low latency, and the original demosaicking algorithm encounters performance bottlenecks. A real-time demosaicking algorithm based on the field programmable gate array (FPGA) was proposed, which could accurately extract the local gradient direction of the image and guide the interpolation restoration of the color. The overall algorithm only needed 7 lines of data delay, and fully considered the hardware characteristics of FPGA, and designed the modules such as line buffer, gradient operator and gradient direction interpolation to reduce the hardware cost. The experimental results show that the algorithm can achieve the micron-level latency, and maintain the restoration effect of texture detail area of the image. The average peak signal to noise ratio (PSNR) on Kodak data set can reach to 38.26 dB.

**Key words:** Bayer array; demosaicking algorithm; FPGA; real-time image processing

## 引言

目前绝大多数的消费级图像采集设备采用

CMOS/CCD 作为前端传感器, 通过 A/D 模块将采集到的模拟量转化为数字信号。由于彩色图像需

收稿日期:2021-11-01; 修回日期:2021-12-01

基金项目:国家自然科学基金 (61872019, 62002005, 6200071402)

作者简介:张泽宇 (1990—), 男, 博士, 主要从事模式识别与图像处理方面的研究。E-mail: zzybeihang@buaa.edu.cn

要红、绿、蓝 3 颜色分量的信息,出于对成本和曝光时间等考虑,消费级产品多采用彩色滤波阵列 CFA(color filter array)的方法,将每个像素点的红、绿、蓝颜色分量压缩,仅保留单个颜色通道信息。其中 Bayer 阵列<sup>[1]</sup>是最常见的一种彩色滤波阵列,其特点在于阵列的绿色分量 2 倍于蓝色或红色分量。

后端图像处理系统通过解马赛克算法将 Bayer 阵列复原为红、绿、蓝 3 色阵列,过程如图 1 所示。利用目标位置原有颜色值以及周围位置的其他颜色分量,恢复目标位置缺失的颜色。算法性能影响了图像复原得到的色彩真实性,进而影响了彩色图像的有效分辨率。在解马赛克的过程中不可避免地引入人为设计的插值像素,从而导致颜色与真实场景不相符的情况发生,主要问题包括拉链效应、伪彩色和摩尔纹效应<sup>[2]</sup>3 大类问题。图 2 展示了典型场景下,由 Bayer 解马赛克算法引入的拉链效应(左)、伪彩色(中)和摩尔纹(右)现象。

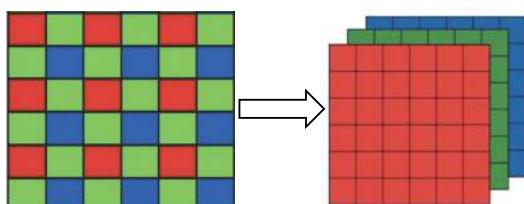


图 1 Bayer 阵列解马赛克示意图

Fig. 1 Schematic diagram of Bayer array demosaic

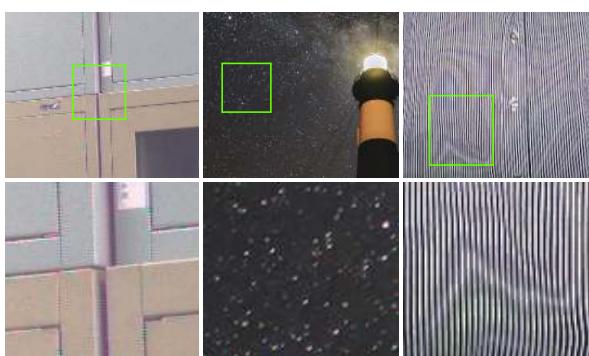


图 2 拉链效应(左)、伪彩色现象(中)及摩尔纹现象(右)示意图

Fig. 2 Schematic diagram of zipper effect, pseudo-color phenomenon and moire phenomenon

目前 Bayer 阵列解马赛克算法已经过了长达 30 余年的发展,现存在的解马赛克算法大致可分为 3 类。第 1 类是模板线性插值方法<sup>[3]</sup>,这种方法的特点在于整幅图像采用相同的插值模板进行计

算,算法复杂度低、输出延迟小。常见的方法包括最近邻插值方法、双线性插值法等。最初的算法设计中,未知的颜色分量仅通过周围的同颜色分量来插值复原,但是由于蓝色、红色分量信息不足,复原效果不佳,引入伪彩色问题。随着色比色差定律<sup>[4]</sup>的提出,算法可以通过更加丰富的绿色信息帮助复原蓝色和红色的颜色信息,这类算法在图像平滑区域实现了很好的效果,但是在图像纹理丰富的区域难以维持锐利的边缘,存在拉链效应等复原问题。

第 2 类是基于局部梯度反向的线性插值方法<sup>[5-7]</sup>,这种算法的提出是为了提升第 1 类方法在图像纹理丰富区域的复原效果。选择垂直于图像梯度的方向进行插值复原,可以有效地减少拉链效应等复原问题。这类算法的性能受制于局部梯度信息的重建精度以及后续的插值方向选择,高效的算法需要权衡梯度算法的复杂度和色彩复原效果。目前广泛应用的是先计算候选颜色,再确定梯度方向的“后验”方法<sup>[6-7]</sup>。该类方法首先计算出不同梯度方向的候选绿色分量,然后利用自然图像中色差沿着图像边缘的方向更加平滑的特点<sup>[8]</sup>确定边缘方向,进一步复原蓝色和红色分量。算法只需要 5~7 行的图像缓存就可以进行瀑布式运算,非常适合基于 FPGA 的开发,适用于实时性要求高的边缘计算场景,如自动驾驶系统中的摄像组件、高灵敏度光电探测设备等。

第 3 类是基于频域分析的解马赛克算法<sup>[9-11]</sup>,该类方法将 Bayer 阵列看作是原始的红、绿、蓝 3 色阵列通过采样得到,在频域中对应特定的频谱调制,而图像中纹理丰富的区域即对应频域中的高频部分。研究人员根据这一特点,设计特定的频域滤波方法解决纹理丰富区域的图像复原效果。方法 [9] 分析了 Bayer 阵列的频域特征,通过设计自适应滤波器的方法来迭代得到最后的红、绿、蓝 3 色阵列。计算过程需要对完整的图像进行缓存,算法延迟时间相较于第 2 类方法慢 2~3 个数量级。该类方法对图像高频区域的复原效果更好,因而适用于专业相片复原精修、视频后期处理等非实时场景的应用。

本文结合实际应用,提出一种低延迟、高复原质量的 Bayer 解马赛克算法,并完成 FPGA 硬件算法的实现。实验结果表明,本文算法可在 FPGA 中实时运行,与相同算法复杂度、低输出延迟的算法

相比, 复原出彩色图像的质量更高。

## 1 系统设计方案

如图 3 所示, 本方案采用 Xilinx 公司的 FPGA 作为主芯片, 实现了将前端 CMOS 采集的图像信息通过数字图像信号处理(ISP)后, 最后通过视频编码方式输出的功能。Bayer 解马赛克算法作为图像信号处理中最重要的环节之一, 部署在 FPGA 内部。本章节重点讲述系统的硬件系统设计方案以及基于 FPGA 的实时数字图像信号处理模块的功能化设计。

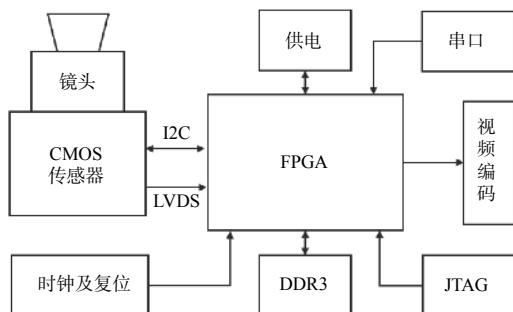


图 3 基于 FPGA 的图像采集与解马赛克系统设计方案  
Fig. 3 Systematic design scheme of image acquisition and demosaic based on FPGA

硬件系统包括供电模块、时钟及复位模块、DDR3 缓存模块、视频编码模块、FPGA 主芯片以及前端 CMOS 芯片。FPGA 作为数字图像信号处理主芯片<sup>[12]</sup>, 不仅完成对 CMOS 芯片和其他模块的配置工作, 同时负责完整的数字图像信号处理功能(包括解马赛克算法的实现); 图像数据缓存采用 DDR3 芯片; 视频编解模块将 FPGA 的输出视频流编码为常见的视频格式。

系统的核心功能都部署在主芯片 FPGA 上, 根据模块的功能不同, 采用 FPGA 内部的 MicroBlaze 软核和 Verilog HDL 硬件语言搭建的功能化模块相结合的方案设计了整体的 FPGA 内部功能框架。MicroBlaze 软核实现了包括 CMOS 芯片寄存器的配置、控制 CMOS 自动曝光和自动增益等功能。模块通过 C 语言实现, 功能特点在于模块参数需要频繁修改、涉及复杂的代数运算以及实时性要求较低等。其他的数字图像处理功能, 包括串行数据解码、坏点补偿、解马赛克模块、Bayer 去噪模块、图像校正等, 均通过 Verilog HDL 硬件语言完成模块化的设计。整体视频数据采用串

行流式的方案, 从 CMOS 的原始输出到最终的制式视频编码完成, 这保证了整体系统具备最低的视频输出延迟。FPGA 功能框架设计方案如图 4 所示。

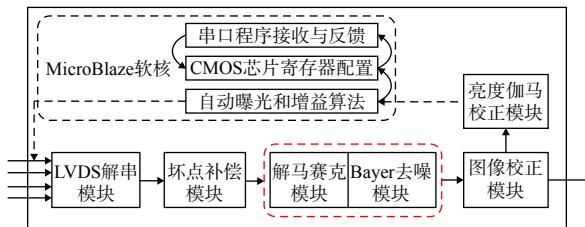


图 4 FPGA 功能框架设计方案  
Fig. 4 Design scheme of functional framework based on FPGA

## 2 实时 Bayer 解马赛克算法的设计与实现

本章重点描述基于 FPGA 的实时 Bayer 解马赛克算法的解压缩过程。本系统采用了改进的基于局部梯度方向的线性插值算法。首先, 算法分别计算出沿水平、垂直方向和平滑的绿色插值作为候选的绿色分量; 在复原了绿色分量信息后, 可以利用自然图像中色差沿着图像边缘的方向更加平滑的特点, 将原有的红色或蓝色分量与候选绿色分量分别计算色差, 色差较小的候选方向即估计为该处的边缘梯度方向; 最后, 根据估计出的局部梯度方向信息, 在梯度的垂直方向上进行插值计算, 复原红色和蓝色分量。

系统根据 FPGA 硬件电路的特点, 设计了 3 大模块, 即行列缓存模块、Bayer 去噪声模块和解马赛克模块。行列缓存模块利用 FPGA 内部 Block RAM 进行数据的行缓存, 通过  $6 \times \text{FIFO}$ (first input first output) 行缓存模块, 得到总共 7 行的图像数据。再利用移位寄存器进行列缓存, 总共利用  $7 \times 10$  的图像矩阵完成算法的功能。Bayer 去噪声模块是为了解决 CMOS 中存在的坏点和噪声点问题, 由于该噪声经过解马赛克算法后会扩散到更大的区域, 因此在解马赛克模块之前进行, 这里采用  $3 \times 3$  中值滤波模块, 具体不进行详述。解马赛克模块完成了包括候选绿色分量的方向插值、局部梯度方向估计和红、蓝色分量复原的工作。整体算法的硬件运算流程如图 5 所示, 流程重点展示了图像数据的瀑式流水处理过程。详细步骤见章节 2.1、2.2 和 2.3。

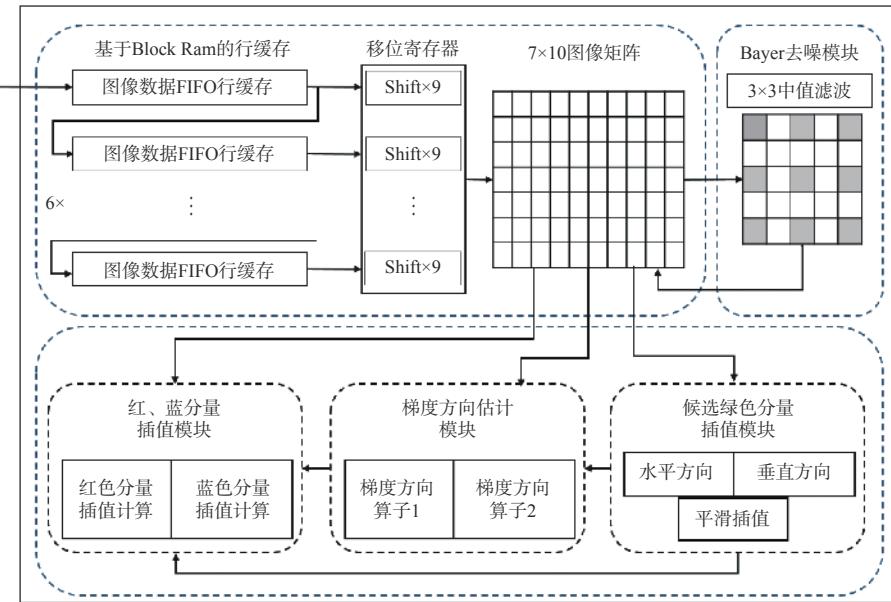


图 5 基于 FPGA 的实时 Bayer 解马赛克算法流程图

Fig. 5 Flow chart of real-time Bayer demosaicking algorithm based on FPGA

## 2.1 候选绿色分量的方向插值方法

候选绿色分量的方向插值方法如图 6 所示。

由于在 Bayer 阵列的排布中, 绿色分量的采样率是红色或蓝色分量的 2 倍, 绿色分量的复原效果要好于红色或蓝色分量<sup>[13]</sup>。因此往往先复原绿色分量, 因为这时还未计算出该位置的梯度方向, 分别计算沿水平、垂直方向以及平滑插值得到候选绿色分量  $G^H$ 、 $G^V$  和  $G^S$ 。

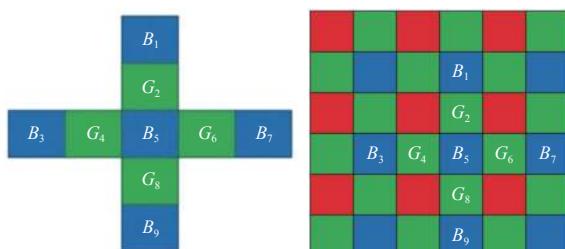


图 6 候选绿色分量插值方法示意图

Fig. 6 Schematic diagram of interpolation method of candidate green components

对于丢失绿色分量的位置而言, 无论该位置原像素是红色还是蓝色, 绿色分量在四周的分布情况是一致的。以当前位置为蓝色像素为例,  $G^H$ 、 $G^V$  和  $G^S$  的计算方法如(1)~(3)式所示:

$$G^H = \frac{(G_4 + G_6)}{2} \quad (1)$$

$$G^V = \frac{(G_2 + G_8)}{2} \quad (2)$$

$$G^S = \frac{(G_2 + G_4 + G_6 + G_8)}{4} \quad (3)$$

## 2.2 局部梯度方向的估计

采用局部梯度方向估计的方法是为了解决单一位置梯度方向估计存在较大误差的问题, 本文为了平衡算法的复杂度和梯度估计精度, 采用 3×3 邻域的梯度信息来估计梯度的方向。综合 3×3 邻域内各像素位置的梯度方向<sup>[14]</sup>, 可以有效地解决图像纹理丰富区域的梯度方向杂乱的问题, 从杂乱的梯度方向中定位到主梯度方向。根据当前位置丢失的颜色分量信息的不同, 需要通过两种不同的梯度算子来计算沿水平和垂直方向的梯度值。局部梯度方向估计的算法流程总结为 4 步:

- 1) 对 3×3 邻域内的每个位置计算水平和垂直方向梯度算子;
- 2) 比较两个方向的算子大小, 用标志位 0 或 1 作标记;
- 3) 统计局部 3×3 邻域内各个位置的标志位, 求和得到总梯度算子;
- 4) 对总梯度算子设置阈值, 判断该位置最可能的梯度方向。

梯度算子 1: 对于丢失绿色分量的位置, 采用梯度算子 1 计算该处的梯度值。以中心位置蓝色像素为例, 如图 7(a)所示, 算子利用蓝色分量的二阶梯度加上绿色分量的一阶梯度分别计算水平和垂直方向的梯度算子值, 通过(4)、(5)式计算出水

平梯度值  $Grad^H$  和垂直梯度值  $Grad^V$ :

$$Grad^H = |G_4 - G_6| + |2B_5 - B_3 - B_7| \quad (4)$$

$$Grad^V = |G_2 - G_8| + |2B_5 - B_1 - B_9| \quad (5)$$

梯度算子 2: 对于丢失红色或蓝色分量的位置, 采用梯度算子 2 计算该处的梯度值, 如图 7(b) 所示, 算子利用绿色分量的一阶梯度和二阶梯度分别计算水平和垂直方向的梯度算子值, 通过(6)、(7)式计算出水平梯度值  $Grad^H$  和垂直梯度值  $Grad^V$ 。

$$Grad^H = |G_1 - G_3| + |G_7 - G_9| + |2G_5 - G_{12} - G_{13}| \quad (6)$$

$$Grad^V = |G_1 - G_7| + |G_3 - G_9| + |2G_5 - G_{10} - G_{11}| \quad (7)$$

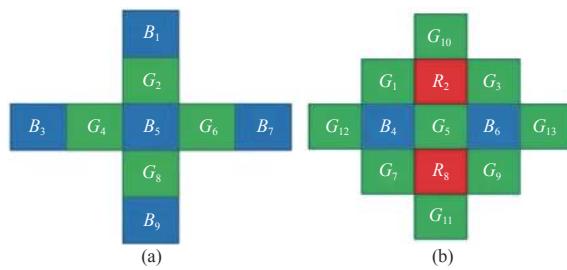


图 7 梯度算子 1、梯度算子 2 模式示意图

Fig. 7 Schematic diagram of gradient operator 1 and gradient operator 2

通过梯度算子 1 和梯度算子 2 计算出  $3 \times 3$  邻域内各个位置的水平和垂直方向梯度值后, 用标志位 Flag 标记各个位置的梯度方向。当  $Grad^H > Grad^V$  时, 判断该位置梯度方向为水平方向, 标志位记为 1; 当  $Grad^H < Grad^V$  时, 判断该位置梯度方向为垂直方向, 标志位记为 0。综合 9 个位置的梯度信息, 按照(8)式估计当前位置的局部梯度信息。

$$\text{Interp} = \begin{cases} \text{horizontal, if } \sum_1^9 \text{Flag}_i > 6 \\ \text{vertical, if } \sum_1^9 \text{Flag}_i < 4 \\ \text{smooth, else} \end{cases} \quad (8)$$

### 2.3 红、蓝色分量的复原方法

经过候选绿色分量和局部梯度方向的计算, 已经具备了完整复原红、绿、蓝 3 层阵列的条件。在对红色和蓝色分量进行插值时, 采用的基本思想是基于色比色差的插值算法。利用色差思想与色比恒定的思想是一致的, 它认为在图片的小平滑区域内, 色差是恒定的<sup>[4]</sup>, 即假设采样点  $(i, j)$  附近的一个采样点  $(m, n)$ , 色差满足(9)式:

$$R_{ij}(B_{ij}) - G_{ij} = R_{mn}(B_{ij}) - G_{mn} \quad (9)$$

本文将插值方法按照采样点颜色分量的不同

分为 4 种模式, 颜色分量排列如图 8(a)~8(d) 所示。当采样点颜色为绿色时, Bayer 阵列的分布如图 8(a)、8(b) 所示, 算法需要插值产生红色和蓝色分量, 差别在于蓝色分量和红色分量在水平和垂直方向的分布不同, 可以采用类似的插值算法; 当采样点颜色为红色或蓝色时, Bayer 阵列的分布如图 8(c)、8(d) 所示, 算法需要插值产生绿色和红(蓝)色分量, 图 8(c)、8(d) 模块差别在于采样点的颜色分量为红色或蓝色, 两种模式下的插值算法策略是一样的。

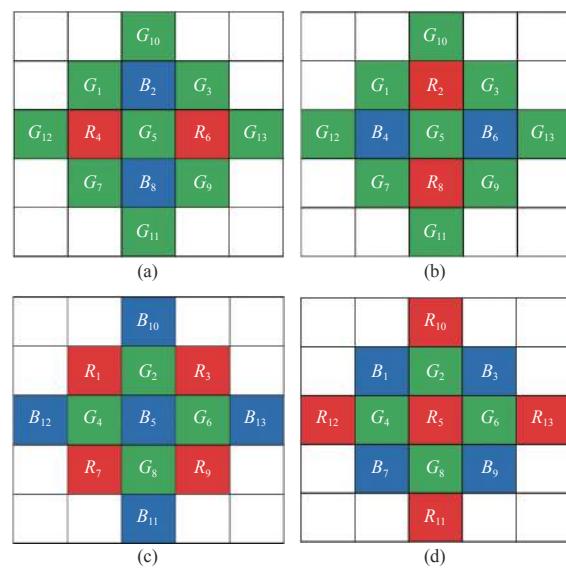


图 8 4 种模式 Bayer 阵列的排列示意图

Fig. 8 Schematic diagram of Bayer array with four patterns

以图 8(a) 模式为例, 利用(10)~(11)式的插值算法来复原红色和蓝色分量。

$$R_5 = \begin{cases} G_5 + \frac{R_4 + R_6}{2} - \frac{G_4^H + G_6^H}{2}, & \text{if Interp = horizontal} \\ G_5 + \frac{R_4 + R_6}{2} - \frac{G_4^V + G_6^V}{2}, & \text{if Interp = vertical} \\ G_5 + \frac{R_4 + R_6}{2} - \frac{G_4^S + G_6^S}{2}, & \text{if Interp = smooth} \end{cases} \quad (10)$$

$$B_5 = \begin{cases} G_5 + \frac{B_2 + B_8}{2} - \frac{G_2^H + G_8^H}{2}, & \text{if Interp = horizontal} \\ G_5 + \frac{B_2 + B_8}{2} - \frac{G_2^V + G_8^V}{2}, & \text{if Interp = vertical} \\ G_5 + \frac{B_2 + B_8}{2} - \frac{G_2^S + G_8^S}{2}, & \text{if Interp = smooth} \end{cases} \quad (11)$$

以图 8(c)模式为例, 利用(12)~(13)式的插值算法来复原红色和绿色分量。

$$G_5 = \begin{cases} \frac{B_5}{2} + \frac{G_4 + G_6}{2} - \frac{B_{12} + B_{13}}{4}, & \text{if Interp = horizontal} \\ \frac{B_5}{2} + \frac{G_2 + G_8}{2} - \frac{B_{10} + B_{11}}{4}, & \text{if Interp = vertical} \\ \frac{G_2 + G_4 + G_6 + G_8}{4}, & \text{if Interp = smooth} \end{cases} \quad (12)$$

$$R_5 = \begin{cases} G_5^H + \frac{R_1 + R_3 + R_7 + R_8}{4} - \frac{G_1^H + G_3^H + G_7^H + G_9^H}{4}, & \text{if Interp = horizontal} \\ G_5^V + \frac{R_1 + R_3 + R_7 + R_8}{4} - \frac{G_1^V + G_3^V + G_7^V + G_9^V}{4}, & \text{if Interp = vertical} \\ \frac{R_1 + R_3 + R_7 + R_9}{4}, & \text{if Interp = smooth} \end{cases} \quad (13)$$

### 3 实验结果分析

为了对本文的解马赛克算法效果进行准确评

价, 我们在广泛使用的柯达图片数据集上进行了定量测量。该数据集包含 24 张真彩色图片, 如图 9 所示。实验过程中采用硬件仿真的方法, 以保证真实的实验效果。将原始柯达图片通过 Matlab 程序采样为 Bayer 阵列, 通过 ModelSim 工具读取 RAW 格式的 Bayer 图片, 经 FPGA 综合生成的解马赛克算法模块后输出彩色的 RAW 数据, 再用 Matlab 工具将 RAW 格式数据转换成 TIF 格式图片, 进行后续的指标评价处理。

本文采用了评价去马赛克重建图像质量最常用的指标峰值信噪比(peak signal-to-noise ratio, PSNR), 定义如(14)~(15)式。

$$\text{PSNR} = 10 \lg \left( \frac{255^2}{\text{MSE}} \right) \quad (14)$$

$$\text{MSE} = \frac{1}{HW} \sum_{j=1}^H \sum_{i=1}^W (I_o(i, j) - I_d(i, j))^2 \quad (15)$$

式中:  $I_o$  和  $I_d$  分别代表原始图像和解马赛克后生成的图像; 图像宽度为  $W$  像素, 高度为  $H$  像素。PSNR 的值越大代表算法复原出的图像质量越高。



图 9 柯达数据集

Fig. 9 Kodak data set

实验对比了双线性插值算法、实时彩色复原算法<sup>[15]</sup>和本文算法, 表 1 展示了代表性图片 01.tif、

07.tif、19.tif 和 24.tif 图片上算法复原图片的 PSNR 结果以及数据集上的平均 PSNR 值。

表 1 3 种算法在柯达数据集上的 PSNR 值

Table 1 PSNR of three algorithms on Kodak data set

dB

图片	双线性插值结果			文献[15]结果			本文结果		
	红	绿	蓝	红	绿	蓝	红	绿	蓝
01.tif	24.833	29.464	25.201	28.764	35.576	28.677	<b>30.121</b>	<b>38.131</b>	<b>29.899</b>
07.tif	31.911	36.231	28.263	36.261	41.942	35.412	<b>37.133</b>	<b>42.576</b>	<b>36.673</b>
19.tif	26.653	31.721	26.262	30.311	37.241	30.276	<b>37.341</b>	<b>41.543</b>	<b>37.015</b>
24.tif	26.164	29.512	25.112	29.612	<b>34.863</b>	<b>28.878</b>	<b>30.141</b>	33.697	28.671
平均值	<b>30.71</b>			<b>34.46</b>			<b>38.26</b>		

对数据集中的“灯塔”图片进行了定性的分析,图 10 展示了经典场景灯塔下的 Bayer 解马赛克效果。其中,图左侧展示的是全图效果,右侧小图是放大的局部细节效果。可以看出,本文提出的方法在纹理复杂的栅栏区域色彩复原效果有了很大程度的改善,伪彩色和拉链效应的现象得到了明显的抑制。



图 10 灯塔图片的解马赛克算法效果对比图

Fig. 10 Comparison of effect of demosaicking algorithm of lighthouse images

除此之外,本系统还针对实际应用中的夜晚星空场景进行了测试,结果如图 11 所示。其中,左侧是局部放大的图像复原效果,右侧是整幅图像的复原图。可以看出,本文提出的方法在复杂的星空背景下,有效地减小了解码过程中引入的伪彩色噪声,保证了星点位置采集的有效性。

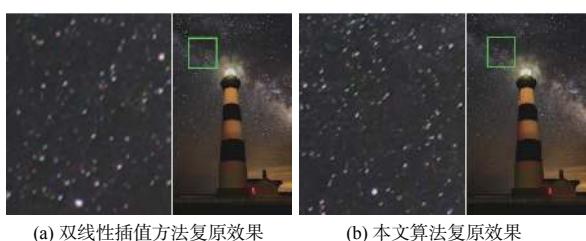


图 11 星空场景下的解马赛克算法效果示意图

Fig. 11 Schematic diagram of effect of demosaicking algorithm in starry sky scene

本文算法运行在 Xilinx 公司的 XC7A50T 硬件平台上,在输入视频图像的分辨率和帧频为

1 080P@30 Hz、像素位宽为 10 bit 的情况下,试验测试了 Bayer 解马赛克模块在 FPGA 上的资源占用率和实际运行速度,资源占用结果如表 2 所示,运行时间为 207.2 μs。

表 2 FPGA 资源占用率

Table 2 FPGA resource occupancy rate

资源名称	FF	LUT	BRAM	DSP	BUFG	MMCM/PLL
可用量/个	65 200	32 600	75	120	32	10
已用量/个	2 737	3 005	10	18	2	1

## 4 结论

本文介绍了一种利用局部梯度方向进行 Bayer 解马赛克的算法及其硬件实现。首先,对算法运行的硬件平台做了简要的概述,展示了算法在数字图像处理系统中的定位和意义;其次,针对 FPGA 硬件平台的特点,阐述了解马赛克算法在 FPGA 中的模块化硬件设计方案和数据流走向,利用特定的数据流水和硬件架构设计有效地降低了硬件开销,保证了系统的低延迟性能;最后,详细地介绍了算法的程序流程和具体实现,通过局部特征的精确提取保证了图像复原的效果。实验结果表明,本文提出的解马赛克系统在保证微秒级视频输出延迟的条件下,提升了对图像复杂纹理区域(如密集的条纹状或点状纹理区域等等)的复原效果,保证了图像的有效分辨率,对于实时高清视频采集系统的搭建有重要的应用价值。

## 参考文献:

- [1] BAYER B E. Color imaging array: US, 3971065[P]. 1976-07-20.
- [2] MENON D, CALVAGNO G. Color image demosaicking: an overview, signal Processing[J]. Image Communication, 2011, 26(8-9): 518-533.
- [3] 冈萨雷斯. 数字图像处理[M]. 阮秋琦, 阮宇智, 译. 北京: 电子工业出版社, 2006: 224-226.
- GONZALEZ. Digital image processing [M]. Translated by RUAN Qiuqi, RUAN Yuzhi. Beijing: Electronic Industry Press, 2006: 224-226.
- [4] 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.
- [5] 杨晓飞, 田启川, 杨余. 基于多梯度的区域自适应去马

- 赛克算法[J]. 计算机应用研究, 2011, 28(12): 4766-4769.
- YANG Xiaofei, TIAN Qichuan, YANG Yu , et al. Adaptive region demosaicking using multi-color gradient[J]. Application Research of Computers 2011, 28(12): 4766-4769.
- [ 6 ] WANG X, LIN W, XUE P. Demosaicing with improved edge direction detection[J]. IEEE International Symposium on Circuits and Systems, Singapore, 2005, 3: 2048-2050.
- [ 7 ] GUNTURK B K, GLOTZBACH J, ALTUNBASK Y, et al. Demosaicing: color filter array interpolation[J]. IEEE Signal Processing Magazine, 2005, 22(1): 4454.
- [ 8 ] 刘方. 基于Bayer彩色滤波阵列插值算法的研究[D]. 成都: 电子科技大学, 2006.
- LIU Fang. Research based on color filter array interpolation algorithm of Bayer pattern[D]. Chengdu: University of Electronic Science and technology, 2006.
- [ 9 ] DUBOIS E. Frequency-domain methods for demosaicking of Bayer-sampled color images[J]. IEEE Signal Processing Letters, 2005, 12(12): 847-850.
- [ 10 ] LIAN N X, CHANG L, TAN Y P, et al. Adaptive filtering for color filter array demosaicking[J]. IEEE Transactions on Image Processing, 2007, 16(10): 2515-2525.
- [ 11 ] CHUNG K H, CHAN Y H. Color demosaicing using variance of color differences[J]. IEEE Transactions on Image Processing, 2006, 15(10): 2944-2955.
- [ 12 ] 叶懋, 黄品高, 吕洋, 等. 基于FPGA的雾霾视频图像实时复原系统研究[J]. 应用光学, 2019, 40(5): 812-817.
- YE Mao, HUANG Pingao, LYV Yang, et al. Research on FPGA-based real-time video restoration system for smog image[J]. Journal of Applied Optics, 2019, 40(5): 812-817.
- [ 13 ] 彭晔. 基于CCD图像的彩色插值算法研究[D]. 南京: 南京理工大学, 2007.
- PENG Ye. Research on color interpolation algorithm based on CCD image [D]. Nanjing : Nanjing University of Science and Technology, 2007.
- [ 14 ] HORE A, ZIOU D. An edge-sensing generic demosaicing algorithm with application to image resampling[J]. IEEE Transactions on Image Processing, 2021(20), 11: 3136-3150.
- [ 15 ] 罗潇, 孙海江, 陈秋萍, 等. Bayer格式图像的实时彩色复原[J]. 中国光学与应用光学, 2010, 3(2): 182-187.
- LUO Xiao, SUN Haijiang, CHEN Qiuping, et al. Real-time color restoration of Bayer format image[J]. China Optics and Applied Optics, 2010, 3(2): 182-187.