

引用格式: 田宗浩, 刘 桢, 陈 栋. 基于 HLS 的弹载相机坏点检测校正模块设计[J]. 微电子学与计算机, 2021, 38(2): 56-61.
[TIAN Z H, LIU Z, CHEN D. Design of detection and correction of dead pixels of the ammunition camera based on HLS[J].
Microelectronics & Computer, 2021, 38(2): 56-61.]

基于 HLS 的弹载相机坏点检测校正模块设计

田宗浩, 刘 桢, 陈 栋

(陆军炮兵防空兵学院, 安徽 合肥 230031)

摘 要: 为减少弹载图像传感器在生产、储存以及工作过程中产生的坏点, 本文分析了坏点的成因和固有特性, 提出了一种改进 FAST 角点检测的坏点检测校正算法. 算法通过降低检测核半径和设定动态判断阈值等方法提高坏点检测的准确率, 减少角点误检数量, 并提高对连续坏点及边缘坏点的适应性, 能在校正坏点的同时保留图像中的角点和边缘特征. 利用 Vivado HLS 高层次综合工具, 实现坏点检测校正算法的并行流水线设计, 使资源消耗和时序达到最优; 将综合生成的坏点检测校正 IP 核移植到 PL 端实现, 实现弹载相机坏点的实时校正.

关键词: 弹载相机; 坏点; FAST 算法; HLS

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2021)02—0056—06

Design of detection and correction of dead pixels of the ammunition camera based on HLS

TIAN Zong-hao, LIU Zhen, CHEN Dong

(Army Academy of Artillery and Air Defense, Hefei 230031, Anhui, China)

Abstract: In order to reduce the bad points during the production, storage and operation of the image sensor, this paper analyzes the causes and inherent characteristics of the dead pixels, and proposes an improved FAST corner point detection for dead pixels correction. The algorithm can improve the accuracy of dead pixels detection by reducing the detection radius and setting dynamic judgment threshold, which can reduce the number of corner point misdetection, and improve the adaptability to continuous dead pixels and edge dead pixels. It can retain the corner point and edge features of the image while correcting the dead pixels. Vivado HLS tool is used to realize the parallel pipeline design of the algorithm, and achieve the optimal resource consumption and timing. The IP core of the dead pixels detection and correction is transplanted to the PL terminal to realize the real-time correction of the dead pixels of the missile camera.

Key words: Missile camera; Dead pixels; Features from accelerated segment test; High level synthesis

1 引言

图像制导炮弹^[1]利用图像传感器采集目标反射或辐射的可见光信息形成图像, 并依据图像包含的信息完成目标识别、定位和跟踪. 弹载相机的感光芯片 CMOS 为图像制导炮弹的核心部件, 受 CMOS

制造工艺、存储环境以及光电信号转换错误影响, 图像中不可避免的出现一些错误的像素点, 即坏点 (Dead Pixel)^[2]. 坏点导致图像不能正确感知外界场景, 影响弹载相机曝光控制和白平衡等模块的处理效果, 降低目标检测、定位精度, 其造成的信息损失在硬件上不可逆转, 只能通过算法进行校正处理.

弹载相机图像采集数据量大,回传至地面站处理过程中数据传输(上行/下行)存在较大时延,满足不了系统实时性要求.为此,利用FPGA在并行和流水线设计上的优势对图像进行弹上自主处理,提出了一种适用于弹载相机的快速坏点检测及校正算法,并利用Xilinx研发的HLS(High Level Synthesis)工具^[3]实现算法的FPGA开发,降低软件算法向硬件平台移植难度,完成弹载相机坏点检测校正嵌入式设计.

2 坏点检测及校正算法

2.1 坏点分类

弹载相机存在的坏点可以根据坏点的成因分为静态坏点和动态坏点,动态坏点又可以分为不可逆动态坏点和可逆动态坏点两类.由图像传感器制造工艺造成的器质性损坏出现的坏点为静态坏点,这些坏点在图像传感器中的位置不变;而由弹载相机存储或强震动工作状态造成的像元损坏出现的坏点为不可逆动态坏点,这些坏点在图像传感器的位置随机出现,且不可逆;另外,由光电转换产生的像素值错误为可逆坏点,这些坏点随机出现,对应像元的感光特性没有损失,在下一帧图像采集过程中仍可正常成像.

图像传感器正常工作模式下像素值不会发生阶跃性变化,可以依据像元是否正常工作将坏点分为:亮点、暗点和色点三种.其中,亮点为像元能感光,但是光电转换后输出恒定最大值;暗点为像元损失感光特性,对应像元不能实现光电转换,始终输出0值;色点为像素值异常点,其光电转换关系与领域内的其它像元差别较大,一般将其按照亮点或暗点进行简化处理,降低算法的复杂度.

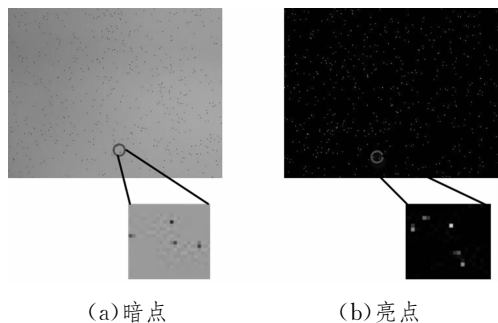


图1 坏点

其中,图1(a)为均匀光源下图像传感器暗点标定图,图1(b)为暗室内图像传感器亮点标定图,暗点和亮点的集合即为图像传感器存在的坏点(不考虑可逆动态坏点).分析坏点周围邻域内像素值可以

发现,正常情况下,中心位置像素值在周围像素值 δ 范围内浮动(图2(a)所示),而当中心像元为坏点时,其与周围像素值之间发生阶跃性变化,即亮点和暗点两种情况(图2(b)和图2(c)).

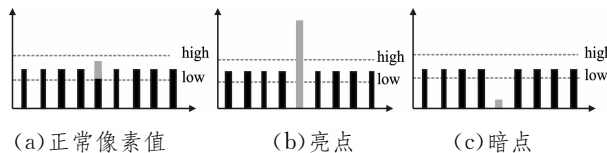


图2 坏点像素值分类

2.2 弹载图像传感器坏点检测校正算法分析

2.1节对弹载图像传感器产生的坏点进行了简单划分,依据坏点产生的原理可知,每个图像传感器出厂之前都应进行静态坏点检测并记录坏点位置,但对每个图像传感器进行坏点检测成本较高,大多数制造商并没有给用户提供图像传感器的静态坏点表.另外,弹载相机在存储和恶劣的工作环境下不可避免的产生新的坏点,这些坏点位置随机出现,固定的静态坏点表不能消除后出现的坏点,并且弹载相机工作平台空间小,弹上处理器存储单元有限,硬件设计过程中存储静态坏点表需要消耗大量的存储资源,影响其它处理算法的设计.为此,笔者结合弹载平台的特殊性以及坏点的属性建立弹载图像坏点实时检测及校正算法,实现对静态和动态坏点的全面检测、校正.

一般来说,坏点独立存在,并与周围邻域内的像素值存在很大的变化,文献[4]提出利用局部像素点的梯度变化检测坏点,并用周围邻域像素值的均值进行坏点校正,该方法在校正坏点的同时也损失了图像大量的角点和边缘特征;此外,不同的滤波算法^[5-6]被用于坏点检测校正,其中非线性空间滤波算法能一定程度上减少图像边缘损失,但是对处于边缘的坏点以及连续坏点处理效果不佳,漏检率高;文献[7]提出了一种多指标坏点检测校正算法,分别对孤立坏点、边缘坏点和连续坏点进行校正,边缘损失小,漏检率低.但算法需要复杂的调参过程,并且需要对待检测点周围领域的梯度变化趋势进行计算,算法复杂度高,时效性较差,硬件移植难度较大.

与角点的特性类似,二维图像亮度变化剧烈的点或图像边缘曲线上曲率极大值的点被定义为角点^[8].在目前广泛应用的角点算法中,FAST(Features From Accelerated Segment Test)角点检测算法对局部特征点的检测速度比传统的SIFT、SURF、GLOH、SURF算法时间开销小很多,准确率高,并且算法简单,但是算法对噪声敏感.也正因如

此,笔者充分分析 FAST 角点检测^[9-10]原理,并结合坏点的特点对其进行改进,实现坏点的准确检测。

传统 FAST 算法以待检测点 P 为圆心,建立邻域半径为 3 的 Bresenham 圆,通过统计分析圆周上像素与中心点 P 的像素差值判断 P 是否为角点,当超过四分之三的连续像素满足式(1)时, P 点为角点,如图 3 所示。

$$|h_p - h_i| > h_{\text{threshold}} \quad (1)$$

式中, h_p 为 P 点的像素值; h_i 为第 i 个像素的像素值; i 为圆周上像素位置; $h_{\text{threshold}}$ 为判断阈值。

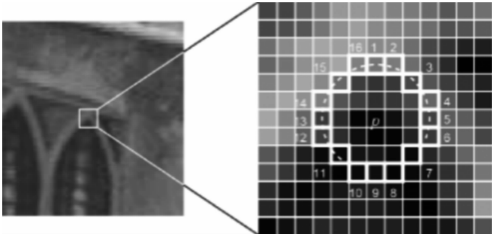


图 3 FAST 算法原理图

直接用上述算法进行坏点检测,不仅检测出图像中的坏点,还会将图像中的角点当成坏点误检。为此,本文以 FAST 算法为基础,改进坏点判断标准,实现校正坏点的同时保存图像的角点信息。

2.3 改进 FAST 算法的坏点检测校正算法

文献[10]提出的 FAST 角点检测算法需要分析检测点 P 与半径为 3 的 Bresenham 圆周上 16 个像素的变化关系,而对 Bresenham 圆内覆盖的 21 个像素的变化情况未充分分析,也正因此如此降低了噪声带来的误检现象,提高了角点检测的准确率。因此,为提高 FAST 算法对坏点的敏感度,本文以待检测点 P 为中心,建立邻域 $\delta=1$ 的检测核,通过分析检测核对角线像素值与待检测点 P 像素值的大小关系确定 P 点是否为坏点,如图 4 所示。

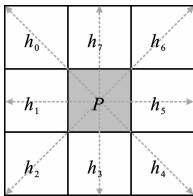


图 4 坏点检测核示意图

图 4 可以分析出,对于待检测点 P 周围的像素来说,对角线位置像素差最能反映检测核内像素的变化趋势。例如,当 P 点为坏点时,其与周围像素的差值大于设定的判定阈值 $d_{\text{threshold}}$ 。另外,对于实时坏点检测来说,判定阈值应为动态值,即不同检测核或

者同一检测核内不同方向判定阈值 $d_{\text{threshold}}$ 不同,这样可以更准确的判定待检测点是否为坏点。为此,本算法将待检测核内某一对称方向像素值差的 n 倍作为判定阈值 $d_{i, \text{threshold}}$,并且当待检测点 P 的像素值与对应像元方向的像素值差值 $d_{i, p}$ 或 $d_{i+4, p}$ 大于判定阈值时, P 点为准坏点,如式 2 所示:

$$d_{i, p} > d_{i, \text{threshold}} \text{ 或 } d_{i+4, p} > d_{i, \text{threshold}} \quad (2)$$

式中, $d_{i, p}$ 为检测点与第 i 个像元像素差, $d_{i, \text{threshold}}$ 为第 i 对称方向判定阈值,满足式 3 所示:

$$\begin{aligned} d_{i, \text{threshold}} &= n \times d_i \\ d_i &= |h_i - h_{i+4}| \\ d_{i, p} &= |h_p - h_i| \\ d_{i+4, p} &= |h_p - h_{i+4}| \end{aligned} \quad (3)$$

式中, d_i 为第 i 个对称像元像素差, h_i 为第 i 个像元像素值, h_p 为待检测点像素值。

为提高本文算法对连续坏点的检测能力,当待检测点周围 8 个像元中有超过 6 个满足式 2 时,判定当前 P 点为一个坏点,并对不满足式 2 的连续坏点进行标定,防止后期出现漏检现象。另外,为降低连续坏点对检测点 P 的校正值带来较大误差,本文将检测点 P 周围不满足式 2 的像元剔除,利用剩余所有邻域像素值的均值 $h_{p, \text{avg}}$ 对坏点 P 进行校正:

$$h_{p, \text{avg}} = \frac{\sum_{i=1}^N h_j}{N} \quad (4)$$

式中, h_j 为满足式 2 像元的像素值; N 为满足式 2 的像素总数; $h_{p, \text{avg}}$ 为坏点校正值。其算法流程如图 5 所示:

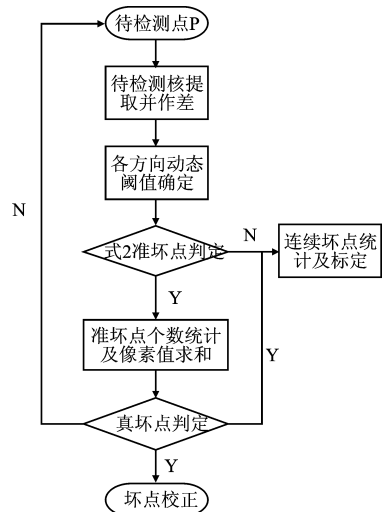


图 5 本文坏点检测校正流程图

为验证本文算法的有效性,分别对比分析文献

[7]提出的坏点检测算法、传统 FAST 算法与本文算法的效果,设置本文算法参数 n 为 2,图 6 和图 7 为各算法坏点检测校正结果,表 1 为各算法坏点检测的统计量:

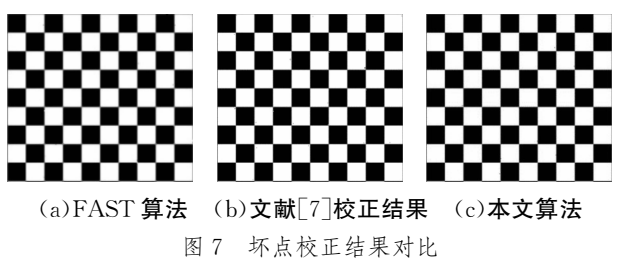
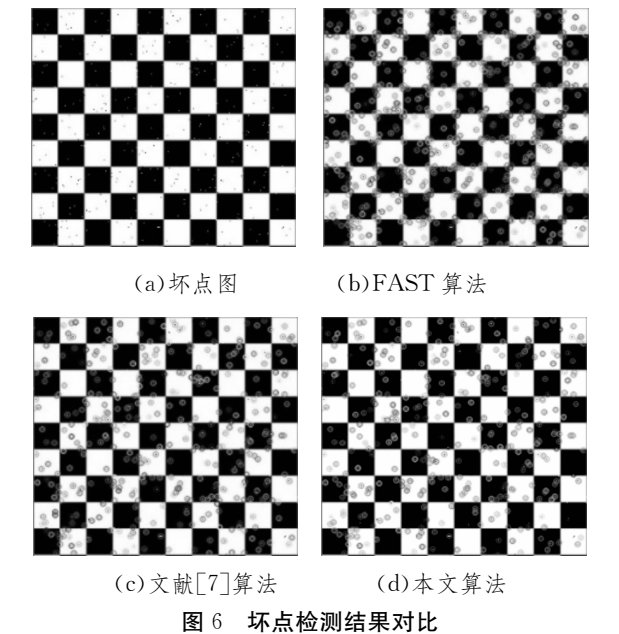


表 1 图像坏点检测数量统计表

算法	坏点数	检测坏点数	漏检数	误检数
FAST 算法	892	1 258	9	375
文献[7]算法	892	924	35	67
本文算法	892	889	7	4

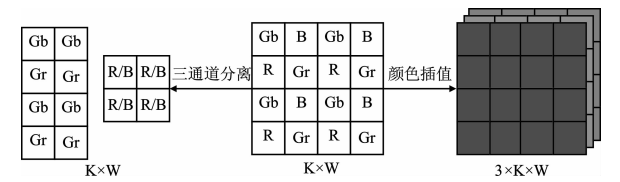
图 6 和图 7 对比分析了三种算法对图像坏点检测校正的结果,传统的 FAST 算法将图像中大部分的坏点和角点同时检测出,误检数较高,校正过程中损失大量的角点信息;文献[7]算法可以检测出图像中大部分坏点,但对边缘坏点,尤其是边缘连续坏点的漏检数较高;本文算法对坏点的检测准确率较高,误检数和漏检数相比其它算法少,并且保边效果较好,对连续坏点的适应性较强.

3 基于 HLS 的坏点检测校正 IP 核设计

HLS 能够利用编译器将 C、C++ 或 SystemC 等高级语言编写的函数转换成 RTL 代码,在系统级的角度进行 FPGA 硬件和软件的协同开发设计,

通过算法优化或者工具指令优化对算法性能进行测试,使其资源消耗和时序达到最优.

本系统图像传感器采集到的原始数据为 Bayer 阵列排布的像素数据,经过颜色插值模块转换成 RGB 或者 YUV 格式,数据量增加 3 倍,如图 8 所示.为此,本算法直接在原始数据基础上进行坏点检测校正,大大降低算法的运算量.



基于 HLS 的图像坏点检测校正过程中,图像数据以 AXI-Stream 的格式传输,通过 AXIvide2Mat 将数据流转换成 HLS::Mat 格式,便于像素的计算和存储.在坏点检测 IP 核设计流程过程中需要进行 for 循环的流水线设计来存储和缓存像素数据,3×3 的待检测核窗口需要一个针对每个像素的 3×3 的数据缓存和至少 2 行像素的数据缓存.即,创建 HLS::window 类型、大小为 3×3 的窗口缓存将数组转化为寄存器组,从而不受 BlockRAM 的访问端口限制;而对于缓存多行像素数据,创建 HLS::LineBuffer 类型、大小为 2 行的数据缓存,实现每次一个像素读写操作,既可以满足流水要求,又节省了寄存器资源.

HLS::window 和 HLS::LineBuffer 存在对数据移动控制接口,直接调用 LineBuffer 的 shift_pixel_up/down 函数需对各行 BlockRAM 进行一次读写操作,再执行 Window 写入会对每个 BlockRAM 再增加一次读操作,使得 Pipeline II 难以做到 1 个时钟周期.为此,将 LineBuffer 的移位操作与 Window 的写入操作同时进行,如图 9(a)虚线框所示:

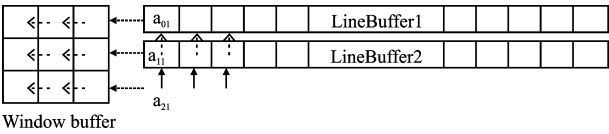
图 9(b)描述了数据流移动规则,实线箭头($\leftarrow$)表示像素流入 Line Buffer 的方向,虚线实心箭头($\leftarrow$)表示各个 Line Buffer 位置像素移动到 window buffer 的方向,虚线单箭头($\leftarrow$)箭头为窗口缓存和行缓存内部数据移动方向,行缓存结束后,当下一个像素 a_{21} 流入时填满窗口缓存(即待检测核),随后在同周期内完成坏点的检测,并输出校正结果.最后,通过综合生成相应的坏点检测校正 IP 核,其资源消耗如图 10 所示.

```
//win column left shift
for(int r= 0;r<KERNEL_SIZE;r++) {
    for(int c=0;c<KERNEL_SIZE-1;c++) {
        win.val[r][c]=win.val[r][c+1];
    }
}

//buf row up shift
win.val[0][KERNEL_SIZE-1] = l_buf.val[0][j];
for(int buf_row=1; buf_row<(KERNEL_SIZE-1); buf_row++)
{
    l_buf.val[buf_row-1][j] = l_buf.val[buf_row][j];
    win.val[buf_row][KERNEL_SIZE-1] = l_buf.val[buf_row][j];
}

//-----
src.data_stream[0] >> srcV;
win.val[KERNEL_SIZE-1][KERNEL_SIZE-1]=srcV;
l_buf.val[KERNEL_SIZE-2][j]=srcV;
}
```

(a) 代码



(b) 数据移动示意图

图 9 数据缓存示意图

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	1
FIFO	6	-	163	593
Instance	12	-	2566	3926
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	-	-
Total	18	0	2729	4520
Available	280	220	106400	53200
Utilization (%)	6	0	2	8

图 10 坏点检测校正 IP 核资源消耗结果图

通过图 10 分析可以看出,坏点检测校正模块通过窗口缓存和行缓存优化处理实现并行流水线设计,使整个模块的硬件资源消耗较低。

4 硬件平台搭建及实验结果

该系统使用 Sony IMX122 传感器进行图像采集,图像大小为1 440×1 080的彩色图像,输入时钟为 27 MHz(148.5 MHz)时,传感器帧频为 30 fps.为满足弹载嵌入式平台小型化和预处理实时性要求,采用 Xilinx 在 28 nm 制造工艺基础上生产的 Zynq-7000 系列的 FPGA.通过合理的运用 FPGA 的可编程逻辑单元 CLB、可编程 I/O 单元和布线资源,达到弹载图像采集及预处理实时性要求。

通过 Xilinx Vivado 开发工具进行硬件平台搭建,将坏点检测校正 IP 核添加到硬件平台中去,实现图像采集、时钟控制、数据转换、坏点检测校正、颜色插值和图像输出功能,其功能结构如图 11 所示:

其中,红色方框为坏点检测校正 IP 核,位于整

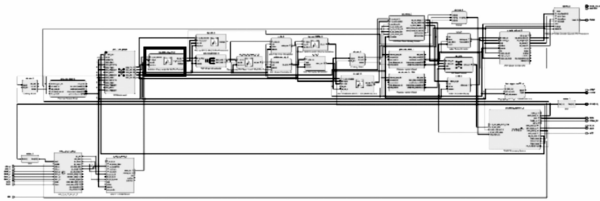
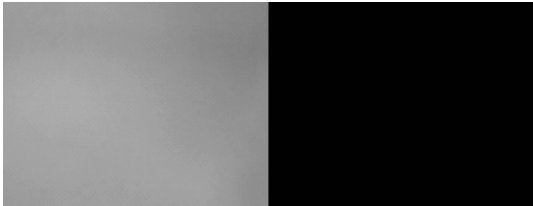


图 11 硬件平台架构图

个 ISP 系统的最前端,图像传感器数据经过 Video In to AXI4-Stream 模块转换为数据流格式,直接进入坏点检测校正 IP 核完成坏点校正,校正后的图像数据通过颜色插值将 Bayer 数据转换为 RGB 格式,并通过 HDMI 输出,其检测校正结果如图 12 所示:



(a) 均匀光源校正 (b) 暗室亮点校正

图 12 坏点检测校正结果

图 12 为采用本文算法搭建的图像采集系统在均匀光源和暗室内的成像结果,与图 1 对比分析发现,经过本文算法坏点校正后图像坏点明显减少,并且在一定程度上降低了图像噪声,获得质量更高的图像。

5 结束语

文章分析了图像传感器坏点产生的成因,并对坏点的固有属性进行总结,在传统 FAST 算法对角点检测的基础上进行算法改进,设定检测坏点的动态判断阈值和限制条件,结果表明改进算法在校正坏点的同时很好的保留了图像的角点特征,坏点检测的误检数和漏检数相对较少,并且对连续坏点有较好的适应性.利用 HLS 开发工具在软件算法向硬件移植的独特优势,高效实现坏点检测算法并行流水线设计.最后在 ZYNQ 平台上实现图像采集、数据转换、坏点检测校正及显示功能,将复杂算法部署在 PL 端,充分发挥 FPGA 的并行计算优势,满足弹载相机低时延、小型化的设计要求。

参考文献:

[1] QIAN L Z. Research on key technology of TV terminal guided artillery Weapon system[D]. Hefei: University of Science and Technology of China, 2006.
[2] XILINX. Xilinx Vivado design suite[EB/OL]. (2017

- 12 — 20). https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug940-vivado-tutorial-embedded-design.pdf.
- [3] DIERICKX B, MEYNANTS G. Missing pixel correction algorithm for image sensors[C]//Proceedings of SPIE 3410, Advanced Focal Plane Arrays and Electronic Cameras II. Zurich, Switzerland: SPIE, 1998: 200-203. DOI: 10.1117/12.324015.
- [4] QIAN F, JIA J J, ZHANG L, et al. Defective pixel correction of spot-detecting camera in satellite-to-ground laser communication ATP system[J]. Chinese Journal of Lasers, 2014, 41(5): 133-139. DOI: 10.3788/CJL201441.0505007.
- [5] CAO J, YUAN Y, SU L J, et al. Adaptive bad pixel correction method for interference-modulated images based on weighted least squares support vector machines (WLS-SVM)[J]. Applied Spectroscopy, 2019, 73(4): 454-463. DOI: 10.1177/0003702819830776.
- [6] LI J S J, RANDHAWA S. Adaptive order-statistics multi-shell filtering for bad pixel correction within CFA demosaicking [C]//Proceedings of 2009 IEEE Region 10 Conference. Singapore: IEEE, 2009: 1-6. DOI: 10.1109/TENCON.2009.5395817.
- [7] HU Y S, HU Y L, YAN M, et al. Detection and correction of dead pixels in an image[J]. Journal of Shanghai University (Natural Science), 2018, 24(5): 755-762. DOI: 10.12066/j.issn.1007-2861.1853.
- [8] ZHU S C, ZHOU D L. Review on image corner detection[J]. Computer Systems & Applications, 2020, 29(1): 22-28. DOI: 10.15888/j.cnki.csa.007237.
- [9] TRAJKOVIC M, HEDLEY M. Fast corner detection [J]. Image and Vision Computing, 1988, 16(2): 75-87. DOI: 10.1016/S0262-8856(97)00056-5.
- [10] HUANG J J, ZHOU G Q, ZHOU X, et al. A new FPGA architecture of FAST and BRIEF algorithm for on-board corner detection and matching[J]. Sensors, 2018, 18(4): 1014. DOI: 10.3390/s18041014.

作者简介:

田宗浩 男, (1991-), 博士. 研究方向为图像处理/智能弹药.

刘 桢(通讯作者) 男, (1981-), 硕士, 副教授. 研究方向为图像处理/嵌入式机器视觉. E-mail: 1109180769@qq.com.

陈 栋 男, (1983-), 博士, 副教授. 研究方向为智能弹药武器系统.

(上接第 55 页)

作者简介:

冯国军 男, (1994-), 硕士研究生. 研究方向为嵌入式系统结构. E-mail: 1214155913@qq.com.

贺占庄 男, (1962-), 研究员, 硕士生导师. 研究方向为计算机系统结构, 嵌入式系统结构.

吕 瑛 男, (1981-), 高级工程师. 研究方向为嵌入式系统结构、物联网无线射频.