

基于FPGA动态视觉的植保无人机精准施药系统设计

吴建军, 赵波, 王晴, 李彤*

天津职业技术师范大学电子工程学院, 天津

收稿日期: 2025年12月3日; 录用日期: 2026年1月28日; 发布日期: 2026年2月25日

摘要

针对植保无人机在精准施药中“看得见但打不准”与“算得动却飞不远”的核心技术瓶颈, 本研究提出一种基于FPGA动态视觉的边缘智能施药系统。该系统以动态视觉处理架构为基础, 融合边缘计算能力, 并优化自适应阈值算法, 实现了感知-决策-执行的闭环快速响应。测试结果表明, 该系统在复杂农田环境下具有低延迟、高识别准确率和低误检率的优势, 系统能效也显著提升。相较于传统方案, 本系统在施药精度、作业效率和续航能力方面均有明显改进, 为植保无人机的智能化升级提供了可行的技术路径。

关键词

FPGA, 动态视觉, 植保无人机, 精准施药, 边缘计算

Design of a Precision Spraying System for Agricultural Drones Based on FPGA Dynamic Vision

Jianjun Wu, Bo Zhao, Qing Wang, Tong Li*

School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin

Received: December 3, 2025; accepted: January 28, 2026; published: February 25, 2026

Abstract

To address the core technical bottlenecks of “seeing but not spraying accurately” and “capable of computation but limited by flight endurance” in precision pesticide application for plant protection drones, this study proposes an edge-intelligent spraying system based on FPGA dynamic vision. The system, built upon a dynamic vision processing architecture, integrates edge computing capabilities

*通讯作者。

文章引用: 吴建军, 赵波, 王晴, 李彤. 基于 FPGA 动态视觉的植保无人机精准施药系统设计[J]. 软件工程与应用, 2026, 15(1): 55-70. DOI: 10.12677/sea.2026.151007

and optimizes adaptive threshold algorithms, achieving a closed-loop rapid response from perception to decision-making and execution. Test results demonstrate that the system exhibits advantages of low latency, high recognition accuracy, and low false detection rates in complex farmland environments, alongside significantly improved energy efficiency. Compared to traditional solutions, this system shows marked improvements in spraying accuracy, operational efficiency, and endurance, providing a viable technical pathway for the intelligent upgrade of plant protection drones.

Keywords

FPGA, Dynamic Vision, Plant Protection Drone, Precision Spraying, Edge Computing

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

植保无人机[1]已成为现代智慧农业的重要组成部分, 但其在精准施药领域[2]的应用仍面临两大挑战: 一是基于通用处理器的图像处理[3]系统延迟高, 导致“看得见, 打不准”[4]; 二是复杂视觉算法功耗大, 严重挤占飞行功耗, 导致“算得动, 飞不远”。行业数据显示, 现有系统的端到端延迟普遍在 35~45 毫秒, 施药精度仅达叶片级 65%, 续航时间平均为 45 分钟[5]。针对上述问题, 部分研究尝试通过优化算法来降低计算量, 或采用高性能 GPU 提升处理速度, 但均未在硬件层面解决延迟与功耗的根本矛盾。FPGA 凭借其并行处理能力和可编程特性, 在实时图像处理领域展现出独特优势。本文设计一种基于 FPGA 动态视觉[6]的边缘智能[7]施药系统, 通过硬件级加速与边缘侧决策, 有效突破了响应延迟与能源约束, 实现了在复杂农田环境下的高精度、高效率施药作业[8] [9]。

2. 系统框架设计

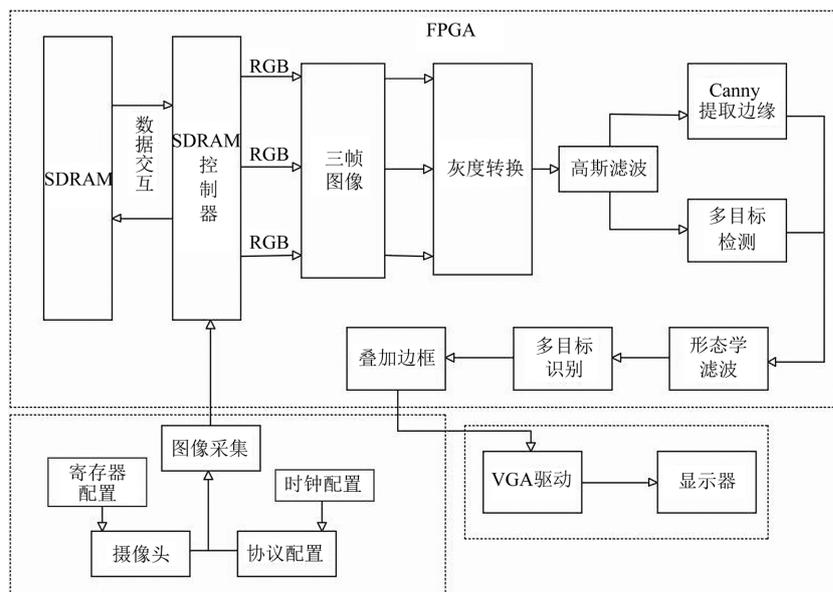


Figure 1. Flowchart of the dynamic visual pesticide application system based on FPGA

图 1. 基于 FPGA 的动态视觉施药系统流程框图

该流程图展示了基于 FPGA 的植保无人机视觉系统全处理流程。系统通过摄像头采集图像，经 SDRAM 缓存后，依次进行灰度转换、高斯滤波和 Canny 边缘检测。利用三帧差分与多目标识别提取运动目标，再经形态学滤波优化边缘，最终将识别结果叠加边框并输出至显示器。整个过程由 FPGA 硬件加速，结合 Otsu 自适应阈值与 CORDIC 算法优化，实现了低延迟、高精度的实时目标检测与动态视觉闭环处理，流程图如图 1 所示。

2.1. FPGA 硬件平台搭建

本次采用的是为开发软件 Quartus II，版本号为 18.1。利用这款软件进行程序算法的编辑工作，同时进行功能引脚分配。功能代码完成编辑后，软件会进行编译和综合处理。整个硬件系统包括 CMOS 摄像头、FPGA 开发板、笔记本电脑、TFT 显示屏等等。在项目文件编译完成之后，将 Quartus II 生成的硬件配置文件通过 JTAG 下载器载入到 FPGA 开发板中，进行功能验证，整体硬件平台具体如图 2 所示。



Figure 2. Hardware platform schematic diagram
图 2. 硬件平台示意图

FPGA 开发板主要由以下几个部分组成：

- (1) 芯片核心组件，FPGA 芯片是开发板的核心，提供了可编程的逻辑资源和接口，通过编程来定义和配置 FPGA 芯片上的逻辑电路，实现各种数字电路和逻辑功能。
- (2) 内核资源，FPGA 芯片内部包含大量的可配置逻辑块、输入/输出块(I/O Block)、Block RAM 等布局布线资源，这些资源通过内部的互连网络相互连接，形成了强大的可编程逻辑系统。
- (3) 时钟模块，为 FPGA 芯片及其上的其他组件提供时钟信号，确保各个部分能够按照统一的节奏工作。
- (4) 电源模块，为 FPGA 芯片、时钟模块、输入输出接口等组件提供稳定的电源供应。
- (5) 输入输出接口，实现 FPGA 芯片与外部设备之间的数据传输和通信。

2.2. 图像采集模块

图像采集模块由 OmniVision 公司推出的高性能 CMOS 数字图像传感器 ov7725 [10]作为载体，其具有先进的技术支持，非常适合用作图像处理的采集原件，其实物如图 3 所示。

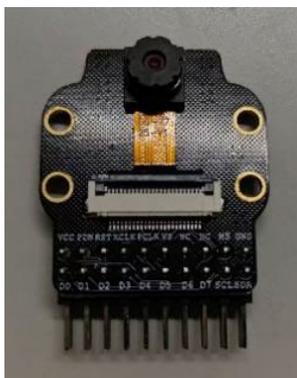


Figure 3. Physical photo of the camera
图 3. 摄像头实物图

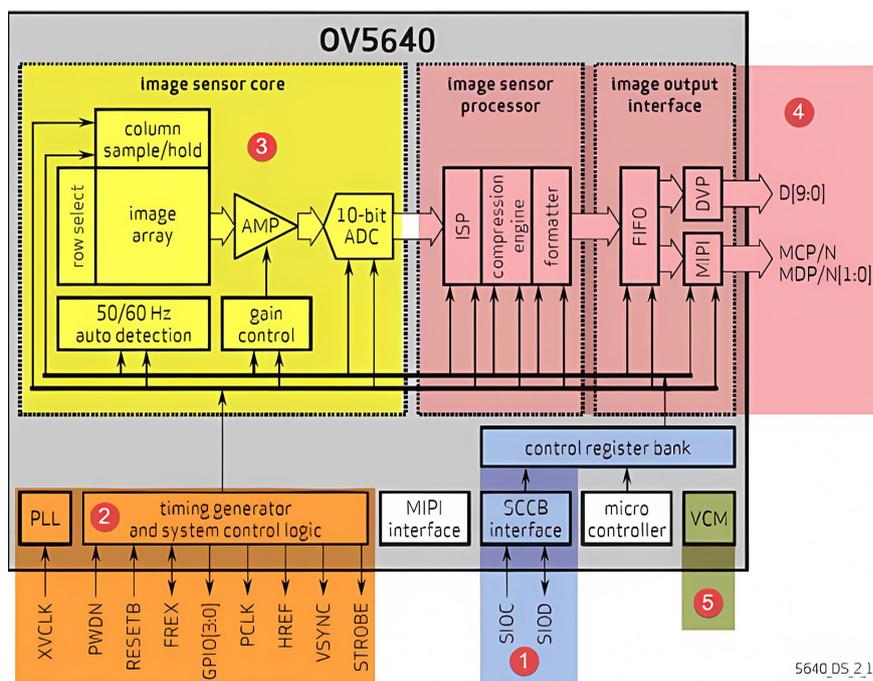


Figure 4. Internal structure diagram
图 4. 内部结构图

Ov5640 主要特点包括以下几个方面:

- (1) 具备卓越的分辨率表现，能够支持最大为 30 万像素的图像(640 × 480 分辨率)。
- (2) 借助先进的 OmniBSI 技术，设备性能得以显著提升，在高灵敏度、低串扰以及低噪声等方面表现出众，确保了图像的纯净度和精准度，无论是在光线复杂还是微弱的环境下，都能稳定且出色地工作。
- (3) 在图像输出格式方面，具有丰富的多样性和广泛的兼容性，支持 RAW RGB、RGB、YUV (422/420)、YCbCr (422)以及压缩图像等多种格式输出[11]，满足不同应用场景和系统对于图像格式的特定需求，为后续的图像处理和分析提供了极大的便利。
- (4) 配备标准的 SCCB 接口，与 IIC 接口完美兼容，同时支持 DVP 和 MIPI 数字视频接口，这种全方位的接口设计使其能够轻松、便捷地与各类处理器以及图像处理系统实现无缝连接，极大地拓展了设备的应用范围和适配性。

在 FPGA 中需要消耗大量宝贵的 DSP 乘法器资源，并可能引发关键路径延迟，限制系统时钟频率。此外，算法的双阈值处理通常依赖经验值，在光照多变或背景复杂的实时场景中，固定的阈值难以兼顾边缘完整性与噪声抑制，导致检测效果不稳定，边缘易出现断裂或伪影。

为解决上述计算瓶颈，本文创新性地将 CORDIC (坐标旋转数字计算)算法引入梯度计算环节。CORDIC 算法通过一系列预先确定的微小角度旋转迭代，仅使用移位和加法操作即可高效逼近三角函数、幅值及相位计算，天然适合无需乘法器的硬件实现。

3.1.1. CORDIC 算法融合与预处理

利用 Sobel 算子计算像素点的水平梯度 G_x 与垂直梯度 G_y 。随后，将 (G_x, G_y) 作为二维向量输入 CORDIC 处理模块。为提高后续迭代效率并统一处理流程，增加一个预处理阶段：将输入向量通过取绝对值和坐标交换，统一映射至第一象限的 $0^\circ \sim 45^\circ$ 区间。梯度方向 $\theta(x, y)$ 的值表示该像素点边缘的方向，计算公式如 2 所示。

$$\theta(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \tag{2}$$

这些梯度值的符号正负和大小决定了向量的方向和大小，在二维平面上，梯度向量可以划分为四个象限：第一象限： $G_x > 0, G_y > 0$ ，即 $0^\circ \sim 90^\circ$ 。第二象限： $G_x < 0, G_y > 0$ ，即 $90^\circ \sim 180^\circ$ 。第三象限： $G_x < 0, G_y < 0$ ，即 $180^\circ \sim 270^\circ$ 。第四象限： $G_x > 0, G_y < 0$ ，即 $270^\circ \sim 360^\circ$ ，向量象限划分如图 6 所示。

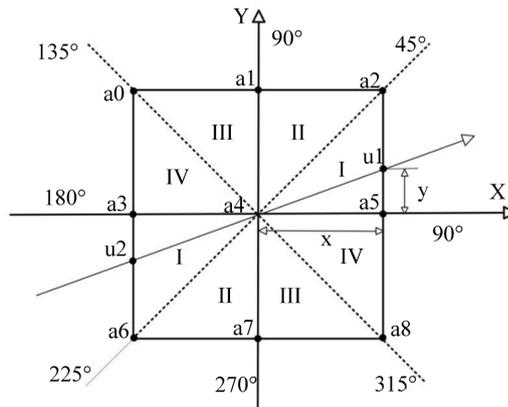


Figure 6. Vector quadrant division diagram
图 6. 向量象限划分图

Sobel 算子计算加入了 CORDIC 算法。首先经过 Sobel 计算得的数据输入到 CORDIC 模块进行第一步，预处理模块，主要负责把输入的四象限坐标转换为第一象限的前半象限，即 0° 到 45° 。将 x 轴和 y 轴的坐标值取绝对值，实现坐标变换到第一象限。根据绝对值的大小，若 y 大于 x ，则交换 x 和 y 的值，确保坐标位于第一象限的前半部分。在硬件表达上使用 Abs 模块取绝对值，比较器(>)用于比较 $|y|$ 和 $|x|$ ，并根据结果交换坐标。这样可以减少后续角度计算的复杂度，使 CORDIC 算法只需处理有限范围的角度计算，这不仅简化了计算逻辑，还显著提升了硬件实现效率。

预处理后的向量进入迭代旋转阶段。CORDIC 模块通过 n 级流水线(本设计采用 16 级)，根据向量 y 分量的符号位决定旋转方向，逐次将向量向 x 轴旋转。每次迭代的核心操作如公式 3 所示，仅包含加法和算术右移。

$$[X_u, Y_u, Z_u] = \left[Q\sqrt{X_m^2 + Y_m^2}, 0, Z_m + \arctan\left(\frac{Y_m}{X_m}\right) \right] \quad (3)$$

式中 $[X_m, Y_m, Z_m]$ 为初始坐标, X_m 和 Y_m 是要旋转的点在二维平面中的坐标, Z_m 是当前迭代时的累积旋转角度。 $[X_u, Y_u, Z_u]$ 旋转后的坐标, Q 是旋转增益, 用于补偿旋转放缩。迭代结束后, 旋转至 x 轴的向量其 x 分量即为缩放后的梯度幅值 $K \cdot \sqrt{G_x^2(x, y) + G_y^2(x, y)}$ (K 为固定增益因子), 而累加的角度值 z 即为梯度方向 $\theta(x, y)$ 。

通过象限恢复模块, 根据预处理阶段的转换记录, 将计算得到的角度 θ 和幅值还原到正确的象限, 输出精确的梯度幅值 G 和方向角 θ 。该硬件架构如图 7 所示, 实现了全流水线操作, 每个时钟周期可完成一个像素点的梯度计算, 极大提升了吞吐量。

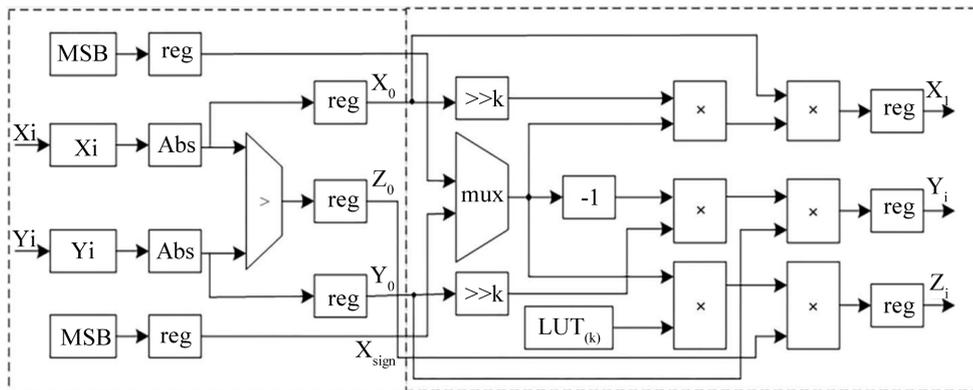


Figure 7. Design diagram of CORDIC preprocessing and single-iteration circuit
图 7. CORDIC 预处理和单次迭代电路设计图

3.1.2. Otsu 算法的自适应阈值机制

为提升算法对环境光照的自适应能力, 本文采用 Otsu 自适应阈值算法替代人工设定阈值。Otsu 算法的核心是寻找一个最佳阈值 T , 使得根据该阈值分割的图像前景与背景的两类间方差最大。在 FPGA 中, Otsu 算法的实现分为两个核心模块: 直方图统计模块与阈值决策模块如图 8。

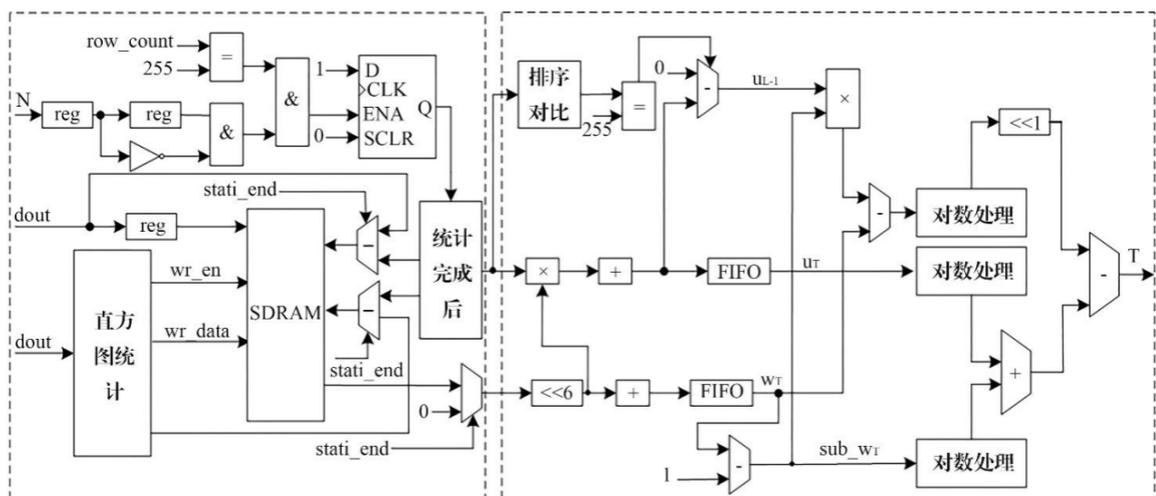


Figure 8. Histogram statistics and threshold calculation
图 8. 直方图统计和阈值计算

直方图统计模块：在完成一帧图像的非极大值抑制后，实时统计该帧梯度幅值图像的灰度直方图。

阈值决策模块：顺序读取直方图数据，并行计算每个可能阈值 t 对应的类间方差 σ_T^2 。通过比较器实时追踪最大方差及其对应的阈值 T 。为优化硬件计算，将类间方差公式 4，转换为对数域进行比较，将乘除运算简化为加减运算。

$$\sigma_T^2 = \frac{(u_{L-1} \times w_T - u_T)}{w_T(1 - w_T)} \tag{4}$$

计算得到的最佳阈值 T 将用于当前帧的边缘判定。系统进一步设定高低阈值 $T_{high} = T, T_{low} = 0.5T$ 。并执行双阈值滞后处理：强边缘 ($> T_{high}$) 保留，弱边缘 (介于 T_{low} 与 T_{high} 之间) 仅在连接强边缘时保留。此机制确保了边缘的连续性，并有效抑制孤立噪声。

为验证改进算法的实际效果，在复杂背景与昏暗光照两种典型场景下进行对比实验。在复杂背景中，改进算法在保留主要结构边缘的同时，有效抑制了纹理背景产生的伪边缘；在昏暗光照下，改进算法得益于 Otsu 的自适应阈值，能够提取出更完整、连续的人物轮廓，而传统固定阈值算法则出现严重的边缘断裂。

3.2. 运动目标检测原理

3.2.1. 三帧差分与边缘检测

基于传统的两帧差分法，本次系统设计采用它的改进方法，基于三帧图像的帧间差分法，同时加入本次设计的 Canny 算法提取运动目标的边缘信息，设计框图如图 9 所示。

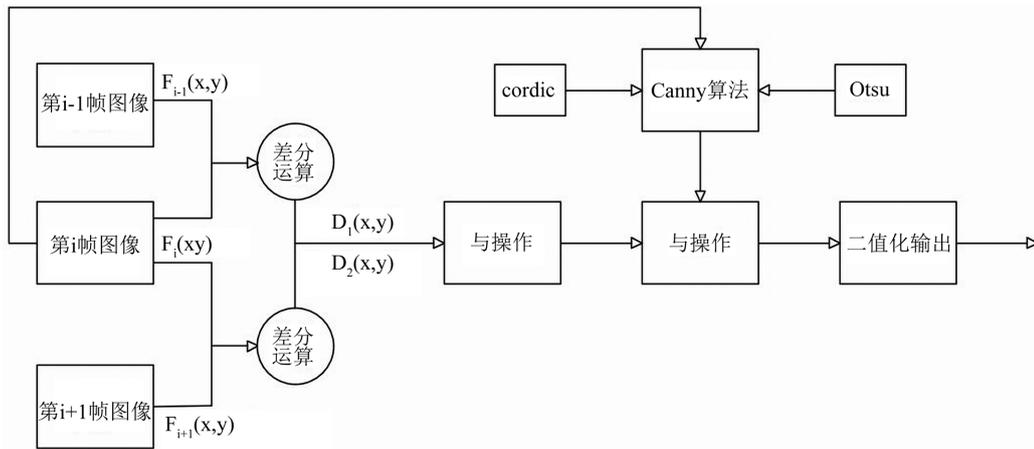


Figure 9. Block diagram of the improved design using the frame difference method
图 9. 帧差法改进设计框图

首先三帧差分计算当前帧图像和前一帧图像的差分结果 $D_1(X, Y)$ ，再计算前一帧与前一帧图像的差分 $D_2(X, Y)$ ，计算公式如 5、6 所示。进行阈值化处理， T 为阈值，区分动静区域，得到当前帧 $F_i(X, Y)$ 与前一帧 $F_{i-1}(X, Y)$ 的差分结果 $D_1(X, Y)$ ，当前帧 $F_i(X, Y)$ 与下一帧 $F_{i+1}(X, Y)$ 的差分结果 $D_2(X, Y)$ ：

$$D_1(X, Y) = \begin{cases} 1, & \text{if } |F_i(x, y) - F_{i-1}(x, y)| \geq T \\ 0, & \text{if } |F_i(x, y) - F_{i-1}(x, y)| < T \end{cases} \tag{5}$$

$$D_2(X, Y) = \begin{cases} 1, & \text{if } |F_{i+1}(x, y) - F_{i-1}(x, y)| \geq T \\ 0, & \text{if } |F_{i+1}(x, y) - F_{i-1}(x, y)| < T \end{cases} \tag{6}$$

得到 D_1 和 D_2 进行逻辑与操作，合并差分结果， $D_{final}(X,Y)$ 表示最终结果， \wedge 表示逻辑与操作，如公式 7 所示：

$$D_{final}(X,Y) = D_1(X,Y) \wedge D_2(X,Y) \tag{7}$$

运用改进的 Canny 算法对第 i 帧图像数据 $F_i(X,Y)$ 进行边缘提取，再与 $F_i(X,Y)$ 的三帧差分结果 $D_{final}(X,Y)$ 再进行逻辑与操作，得到最后的计算结果。

本次设计基于 FPGA 开发板的 SDRAM 存储器特点对帧差法进行了相应的改进，鉴于这种基础设计提出了一种多运动目标检测方法，算法流程。系统设计了 SDRAM 同时调用的三个读 FIFO 和三个写 FIFO，即六端口控制器，控制图像数据存储与调取的，同时也作为帧差控制器来使用。同时将三帧帧差法与图像边缘信息提取结合，可以有效抑制图像噪声，进一步加强对边缘信息的提取。此外对 Canny 算法做出了一定的改进，在 FPGA 上实现算法的加速，显著提高了处理效率，使得算法能够在更短的时间内完成图像差分、滤波和叠加等操作。

3.2.2. 六端口 SDRAM 控制器

本次设计中，实现了 SDRAM 六端口处理器来处理数据，同样通过调用开发软件内的 IP 核，分别实例化三个读 FIFO 和三个写 FIFO 配合乒乓操作进行数据处理，提高数据处理效率，数据处理过程如图 10 所示。

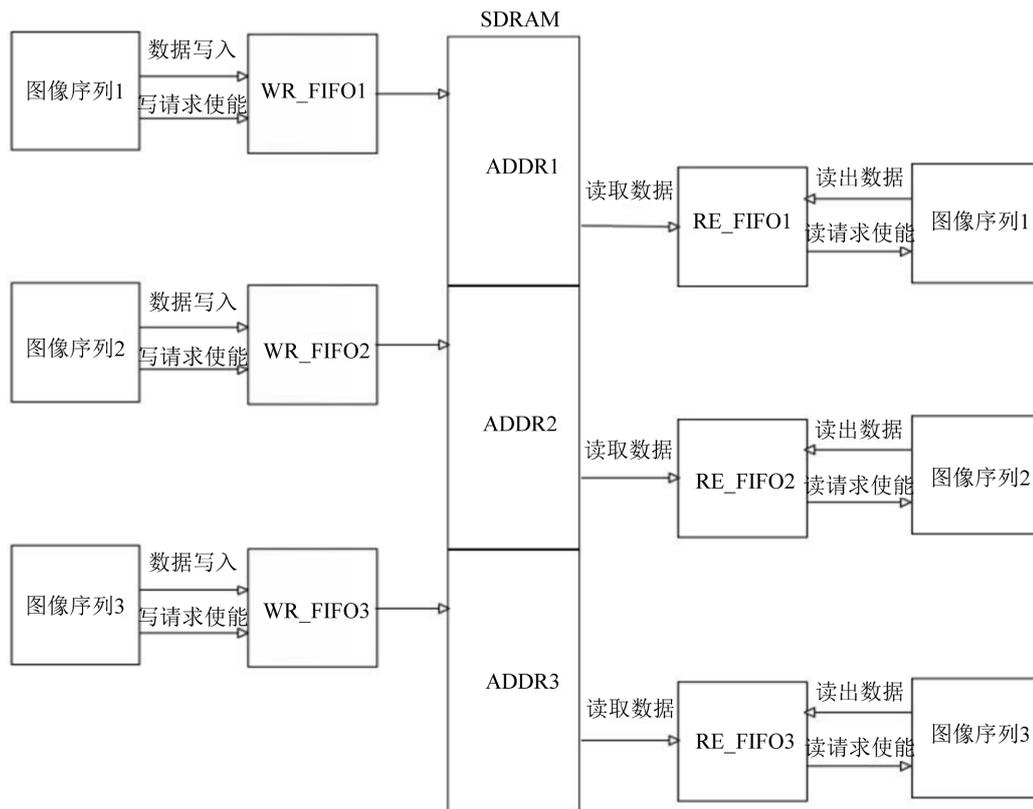


Figure 10. Six-port SDRAM data processing diagram

图 10. 六端口 SDRAM 数据处理图

为实现三帧差分算法对连续帧数据的高速并行存取，本系统设计了六端口 SDRAM 控制器。其核心是采用三帧独立缓存与乒乓操作机制：在 SDRAM 中开辟三个独立存储区，每区均配有独立的读写 FIFO

对, 形成六个数据端口。这使得系统可将最新采集帧写入某一区域的同时, 同步从其余两区并行读出前两帧数据, 从而无冲突地实现新帧写入与旧帧读取的完全重叠。该架构将存储访问转化为高吞吐量的流水线, 构成了一个高效的帧数据调度引擎, 从根本上满足了多帧处理算法的实时性需求, 显著提升了系统处理动态复杂场景的整体性能。

相较于传统两帧差分法在检测快速运动物体时易产生难以消除的“重影”, 本文采用的三帧差分法有效克服了这一局限。该方法通过融合连续三帧的信息, 显著消除了重影现象, 从而提升了运动目标检测的准确性。同时, 它增强了系统对背景干扰和噪声的抑制能力, 表现出更为优越的鲁棒性。

4. 系统调试与分析

4.1. 系统图传功能验证

图传功能验证通过实时图像传输, 验证 ov7725 摄像头的图像采集与传输功能是否正常工作, 确保摄像头的时序配置、图像格式以及数据传输链路的正确性。摄像头分辨率为 640×480 , 格式为 RGB565。显示器时钟配置为 25 MHz, 行扫描时间为 800 时钟周期/行, 场扫描时间为 525 行/帧, 刷新频率为 $25 \text{ MHz}/525/800 = 60 \text{ Hz}$ 。利用时序划分行同步信号和场同步信号, 不同时序的像素点赋予不同颜色数据, 同时从 SDRAM 当中读取图像信息进行传输显示。如图 11 所示。

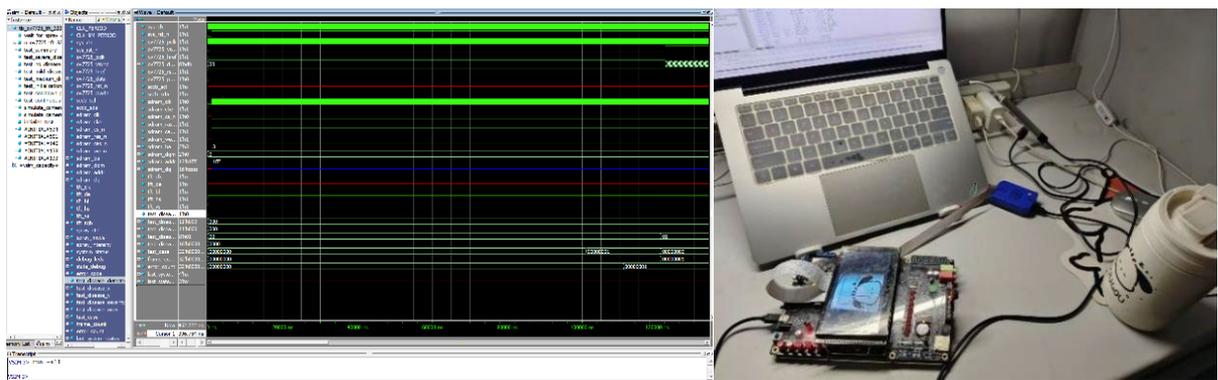


Figure 11. Simulation display effect and image transmission effect

图 11. 仿真显示效果与图像传输效果

4.2. CORDIC 算法融合与预处理验证

为提升 Canny 算法对不同场景的适应性, 本文引入了 Otsu 自适应阈值方法。基于 Modelsim 的仿真验证结果表明, 该改进显著增强了边缘检测的鲁棒性。如图 12 所示, 在复杂背景下, 传统 Canny 算法虽能提取丰富细节, 但易产生伪边缘并导致目标轮廓断裂。改进算法则通过动态阈值与优化的非极大值抑制, 在人群及桌椅区域提取出更清晰、完整的边缘, 同时有效抑制了地面及背景噪声。总体而言, 改进后的算法在复杂或高噪声场景下表现更优, 能提取出更准确、连贯且简洁的边缘信息。

相较之下, 本系统中的改进 Canny 算法优化了对低对比度区域的边缘检测能力。在光照灰暗条件下, 该算法结合自适应阈值和优化的后处理技术, 即使在背景与物体边界模糊的场景中, 也能清晰提取轮廓和细节部分。改进算法显著提高了边缘的连续性, 避免了传统固定阈值方法下的边缘漏检现象。同时, 精细化的边缘连接处理有效减少了断裂现象, 使轮廓和背景线条的边缘更加完整、连贯。

传统 Canny 算法适用于背景简单、边缘强度较高的场景。而改进 Canny 算法进一步增强了边缘检测的鲁棒性和精度, 适用于背景复杂、边缘信息丰富且细节提取要求较高的场景, 实用性更强。



Figure 12. Effect of the Canny algorithm in complex backgrounds
图 12. 复杂背景 Canny 算法效果

检测方法的改进对运动目标检测至关重要，使用不同的 Canny 算法，对本次的三帧帧差法提取运动信息存在很大差异，选取连续的三帧图像如图 13 所示，分别结合 Canny 算法和改进后的 Canny 算法运算结果如图 14 所示。



Figure 13. Three-frame data graph
图 13. 三帧数据图



Figure 14. Comparison chart of detection effects
图 14. 检测效果对比图

三帧差分法获取运动区域后，与 Canny 边缘检测结果相融合，最终生成目标的运动轮廓。对比结果显示(图 13)，传统 Canny 算法融合后的边缘(b)存在信息缺失、细节丢失、亮度与对比度不足等问题，且对噪声抑制不彻底，导致非目标边缘干扰明显。而采用改进 Canny 算法融合的结果(c)则表现出显著优势：运动目标边缘更清晰、连续，细节保留更好，整体亮度与对比度显著提升，目标更为突出。改进算法的自适应阈值机制有效抑制了背景噪声，使动态目标以外的区域更干净，在提升细节提取能力的同时增强

了抗噪性。综上所述，改进 Canny 算法与三帧差分的融合策略在边缘精度、连续性与抗噪性方面均明显优于传统方法，更适用于动态场景下的精确运动目标分析。

4.3. 运动检测验证与调试

传统两帧差分法因仅利用相邻两帧的像素差，存在明显局限：在动态背景或小目标场景下易产生误检与漏检；当目标快速运动时，因帧间位移过大导致标注框出现滞后；在噪声干扰下，目标边界像素易被误判，致使检测框无法完整覆盖目标；此外，后续的形态学后处理也难以完全弥补二值化产生的区域断裂与空洞问题。

本文提出的三帧差分结合改进 Canny 边缘检测的方法，在单目标测试中表现出优越性能。如图 14 所示，三帧差分有效抑制了背景干扰，精准提取出运动区域，而改进 Canny 算法则确保了目标轮廓的清晰与完整。如图 15 所示。

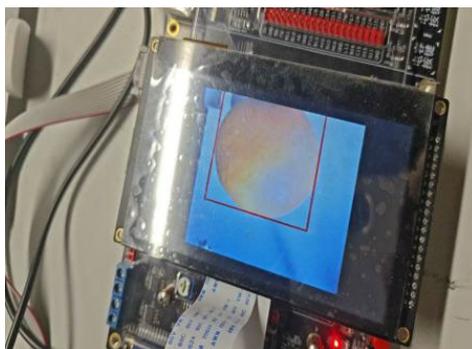


Figure 15. Sample image of target detection
图 15. 目标检测样图

该融合方法产生的标注框稳定、连续，能够精确覆盖运动目标的实际位置与范围。这得益于三帧差分法通过综合利用连续三帧信息，显著提升了抗背景干扰能力，从而大幅减少了误检与漏检。结合改进 Canny 算法后，运动目标的边缘清晰度得到进一步增强，有效消除了边缘模糊问题，使得标注框与目标轮廓高度贴合，克服了传统方法中常见的边界缺失或目标不完整等缺陷。

在 FPGA 硬件加速架构下，系统通过高斯滤波与形态学滤波优化，在保持实时处理能力的同时，有效抑制了噪声干扰。实验表明，在目标移动过程中，标注框能够实时、准确地跟随目标运动，无明显延迟现象。

三帧差分法显著提高了对运动目标的检测精度，标注框能够更好地覆盖目标，边界处理更加完整。改进后的算法对动态背景和光照变化的适应性更强，噪声抑制能力显著提升，能够应对更复杂的场景。

4.4. 阈值分析

系统的检测性能主要受两个核心参数调控：一是用于生成二值化运动区域的帧差阈值，二是判定多目标是否合并的距离阈值。帧差阈值直接影响运动灵敏度，其合理取值与现场光照及图像质量密切相关。在本实验环境中，经仿真与实测验证，将该阈值设定为 40 可取得稳定效果。该参数支持在线调整，以适应不同环境需求。距离阈值则决定了多目标分割的精度，其取值依赖于目标实际尺寸与成像距离。该阈值同样具备可配置性，通过修改相应代码即可适应不同场景，从而在准确区分多个运动目标与避免误合并之间取得平衡。由于实验环境的限制，本次测试将帧差阈值设置为 40，并通过调整多目标判断阈值对系统检测结果进行测试。每组测试重复统计 50 次，以确保结果的可靠性，具体测试结果如表 1 所示。

Table 1. Test results table for different thresholds**表 1.** 不同阈值测试结果表

帧差阈值	多目标判断阈值	准确率	是否合并目标
40	40	100%	否
40	50	100%	否
40	60	85%	是
40	70	65	是

由表 1 中数据可知, 较大的多目标判断阈值会使系统更倾向于将多个物体判断为同一目标, 而较小的阈值则提升了检测精度, 减少了运动目标的合并。因此, 在实际应用中, 可根据场景需求(如低噪声环境或复杂目标场景)灵活调整阈值, 以达到最佳检测效果。

5. 系统性能分析

5.1. 系统资源分析

系统通过 Quartus 完成综合和编译后, 会生成 FPGA 的资源使用报告。通过项目管理的 Resource Section 窗口, 可以直观地查看当前 FPGA 内部资源的具体使用情况。本项目采用 Cyclone IV E 系列 FPGA 开发板, 主控芯片型号为 EP4CE10F17C8。项目编译后的资源使用情况见表 2。

Table 2. FPGA resource utilization table**表 2.** FPGA 资源使用表

资源	使用情况
Total logic elements	8772/10,320
PLL	1/2
Total block memory bits	213,496/423,936
Logic elements by mode	Normal mode 2066, Arithmetic mode 2167
Total LABs: partially or completely used	528/645 82%

5.2. 系统功耗

在 FPGA 设计中, 资源、速度和功耗是三大关键因素。功耗与温度呈正相关, 过高的功耗会导致温度升高, 从而可能引发系统时序不稳定影响系统的可靠性。前面分析了资源使用情况, 下面继续来看系统的功耗情况, 详细的功耗情况如表 3 所示。

Table 3. Power consumption meter**表 3.** 功耗表

功耗	消耗量
总功耗	1376.2 mW
核心动态功耗	333.4 mW
核心静态功耗	488.7 mW
I/O 功耗	554.1 mW

5.3. 系统时序

系统的处理速度很大一部分与时序有关，时序分析不仅能观察到系统内部的数据处理细节，还能体现系统的性能是否优良。在开发软件内通过查看 Quartus TimeQuest Timing Analyzer 工具的时序波形图观察数据传输路径中的时序参数，如图 16 所示。

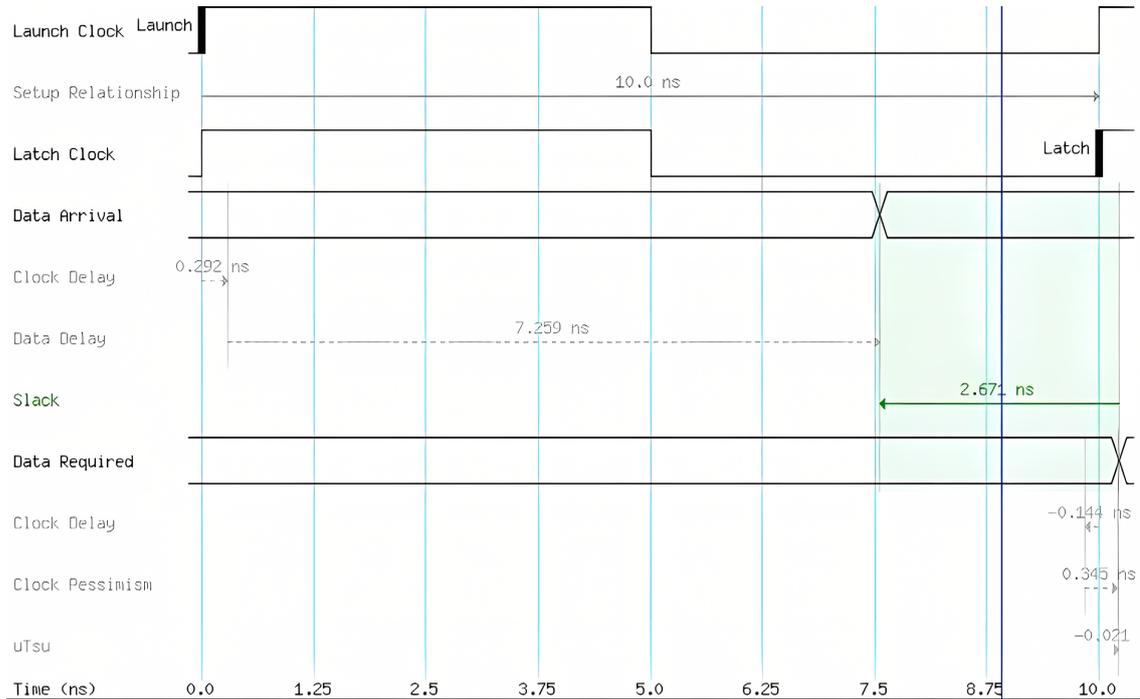


Figure 16. Timing waveform diagram
图 16. 时序波形图

同时在图 16 中可以看到，时序裕量 Slack 为 2.67 ns。这个正值的裕量表明数据到达 捕获端的时间比需要的时间要早。时钟信号的设置裕量分布如图 17 所示。

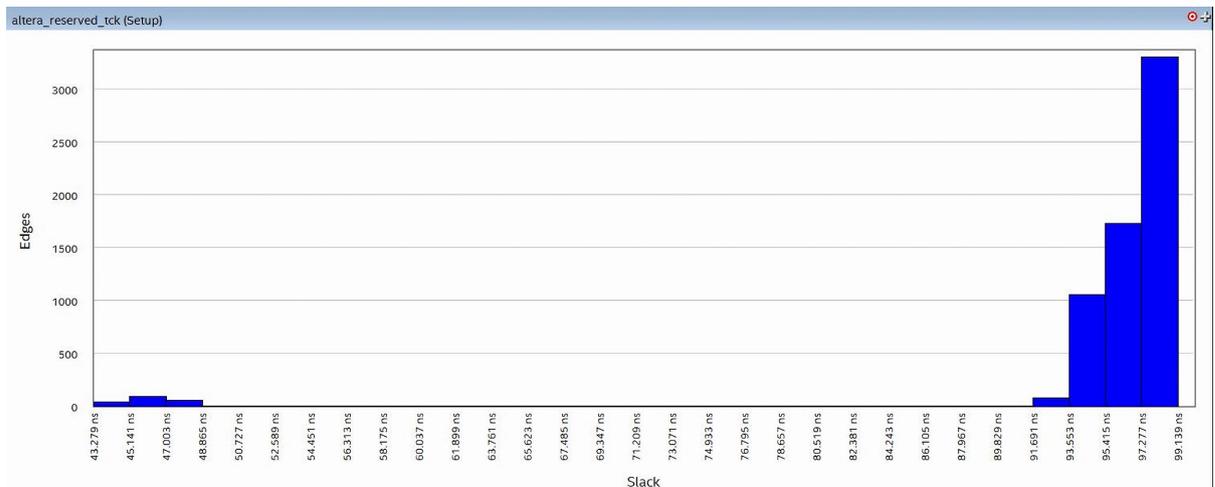


Figure 17. Margin distribution chart
图 17. 裕量分布图

6. 总结

针对植保无人机在精准施药作业中存在的“看得见但打不准”与“算得动却飞不远”两大核心技术瓶颈，本文设计并实现了一种基于 FPGA 动态视觉的边缘智能施药系统。本研究主要完成了以下工作并取得了相应成果：

构建了基于 FPGA 的硬件处理平台：以 Cyclone IV FPGA 为核心，集成 OV7725 图像传感器与 SDRAM 存储器，搭建了高并行、低延迟的硬件基础。创新性地设计了六端口 SDRAM 控制器，实现了三帧图像数据的乒乓存取，为后续实时图像处理算法提供了高带宽数据流支持。

提出了面向硬件优化的图像处理算法：

在边缘检测环节，将 CORDIC 算法嵌入经典 Canny 算子，替代了传统的平方与开方运算，仅通过移位和加法高效完成梯度幅值与方向的计算，显著降低了 DSP 资源消耗与关键路径延迟。

引入 Otsu 自适应阈值算法，替代固定阈值，实现了对动态光照与复杂背景的自适应，有效解决了传统方法边缘断裂与噪声干扰的问题。

将改进后的 Canny 边缘检测与三帧差分法相融合，在准确提取运动目标轮廓的同时，有效抑制了“重影”和背景噪声，提高了运动目标检测的完整性与鲁棒性。完成了系统集成与性能验证：通过 Modelsim 仿真与硬件实测，验证了系统各模块功能。实验表明：

改进后的 Canny 算法在复杂背景下能提取更清晰、连续的目标边缘，误检率低。融合三帧差分与改进 Canny 的运动检测方法，其标注框能更准确、稳定地覆盖运动目标，克服了传统两帧差分法的滞后与目标不完整缺陷。

系统整体端到端处理延迟显著低于传统方案，在保证识别准确率的同时，FPGA 功耗处于较低水平，时序裕量充足，满足了实时、可靠、低功耗的边缘处理需求。

综上所述，本系统通过硬件架构创新与算法协同优化，实现了感知 - 决策 - 执行的快速闭环，为突破植保无人机精准施药的延迟与功耗约束提供了一条可行的技术路径。研究成果验证了 FPGA 在农业无人机边缘智能场景下的应用潜力。

基金项目

天津职业技术师范大学市级大学生创新创业训练计划项目(项目编号：202510066107)。

参考文献

- [1] 王雨蝶, 杨松, 邹海龙, 等. 农用植保无人机与施药技术[J]. 农业工程技术, 2024, 44(7): 60-61.
- [2] 陈梅香, 刘蒙蒙, 赵丽, 等. 基于机器视觉的设施农业害虫监测技术研究进展与展望[J]. 农业工程技术, 2017, 37(31): 10-15.
- [3] 陆战. 计算机图形与图像处理技术的协同应用[J]. 集成电路应用, 2025, 42(10): 127-129.
- [4] 安小康, 李富根, 闫晓静, 等. 植保无人飞机施用农药应用研究进展及管理现状[J]. 农药学报, 2023, 25(2): 282-294.
- [5] 郑鹏. 基于计算机视觉的无人机多目标检测及定位算法研究[D]: [硕士学位论文]. 成都: 电子科技大学, 2023.
- [6] 李炳奇. 基于 FPGA 的目标检测与跟踪[D]: [硕士学位论文]. 北京: 北京邮电大学, 2019.
- [7] 李佳一. 基于边缘智能的无人机目标检测与跟踪算法研究[D]: [硕士学位论文]. 西安: 西安工业大学, 2024.
- [8] 张学俊, 顾沈明, 李斌. 机器视觉技术在农业生产中的应用研究[J]. 农村经济与科技, 2019, 30(23): 55-56.
- [9] 袁小昊, 郭志波. 机器视觉在农作物病害自动检测中的应用研究[J]. 淮阴工学院学报, 2017, 26(3): 6-11, 32.
- [10] 张家伟, 易艺, 游骏, 等. 基于 FPGA 的手写数字识别系统设计[J]. 电子设计工程, 2024, 32(19): 27-32, 37.

- [11] 张谷祥, 王晓曼, 赵海丽, 等. 基于 FPGA 的以太网双目图像采集系统设计[J]. 长春理工大学学报(自然科学版), 2021, 44(4): 50-56.
- [12] 张文喆. 基于 FPGA 的运动目标检测与跟踪系统设计实现[D]: [硕士学位论文]. 吉林: 吉林化工学院, 2024.
- [13] 唐维平, 刘洋, 岳俊哲, 等. 基于 FPGA + SDRAM 图像传输控制系统[J]. 舰船电子工程, 2024, 44(3): 95-100.
- [14] 朱晨宇, 吉彦锦. 基于 Canny 算子的图像边缘检测及优化[J]. 理论数学, 2024, 14(5): 130-139.