

# 基于 FPGA 的视频流白平衡算法硬件加速

许建明 秦杨

(邵阳学院, 湖南 邵阳 422000)

**摘要:** 采用 xc7a100tfgg484-2 芯片设计了用于实时图像采集并色彩矫正的硬件系统, 实现了对彩色 CMOS 相机的自动白平衡处理。针对高速数据流传输时的数据延迟问题, 对传统的白平衡算法在 FPGA 上实现硬件加速, 描述了算法的硬件加速实现思路, 并采用 Verilog 语言在 Vivado2019.2 软件开发平台上完成了自动白平衡算法的程序设计。并通过 Vivado 自带的仿真库进行仿真分析和硬件系统实际效果, 验证了硬件算法的正确性。分析以及实际效果表明该系统可以准确、高清以及低延迟的对视频流进行自动白平衡处理。

**关键词:** FPGA; 图像; 硬件加速

## 引言

当今社会, 图像是信息的主要载体, 图像的采集和处理以及存储已经广泛运用于工业生产、医疗卫生、航空航天等领域。提升图像的处理速度和传输延迟对图像的采集设备有着较大的意义, 如相机、远程手术、工业检测等场合。传统的 ARM 对图像数据的处理采用指令串行的方式, 速度较慢, 对于一些数据量较大、实时性要求较高的图像处理, ARM 就无法满足需求。DSP 受时钟影响, 处理速度有限, 实时性较差。现场可编程逻辑器件 FPGA (Field Programmable Gate Array) 的具有实时性和可重构性, 且能在较高时钟下运行, 能较好的满足实时性要求[1、2、3]。

自然界中, 在不同色温的光源下, 同一物体通常会反射出不同的颜色。但是人脑具有色彩恒常性的相关机制, 可以使同一物体在不同的光源下被识别为近似的颜色, 但是图像传感器本身并不会识别物体本身固有的颜色, 只会将被反射的光记录并传输, 所以在室内钨丝灯光下拍摄的物体就会偏黄, 在日光灯下拍摄的影像会发绿, 而在日光阴影处拍摄的照片会偏蓝。而为了让图像传感器拍摄到的图像更加接近人脑认知的颜色, 则需要引入白平衡调整, 使图像更加还原物体本来的颜色[4]。常用的白平衡算法有灰度世界算法、完美反射算法和动态阈值算法。本文加速的算法为灰度世界算法。

现场可编程逻辑器件 FPGA (Field Programmable Gate Array) 是一种由逻辑单元、RAM、乘法器等硬件资源结合而成的芯片, 具有可重构、定制化、并行执行、无指令集等特点, 用户可以对 FPGA 内部的逻辑块进行重配置, 以实现用户逻辑[5]。相比与 ARM 的一般化, FPGA 的异构更适合在一些对信号处理实时性要求较高的场合。目前 FPGA 已经广泛运用于信号处理、神经网络加速、控制算法、高速接口设计等场合[6]。

## 一、白平衡系统整体框架设计

本次设计的白平衡系统以 FPGA 为核心, 包括了摄

像头驱动、图像处理、缓存和显示模块。

该系统采用 OV5640 摄像头采集图像, 其最大像素为 500 万 (2892 × 1944 分辨率), 输出图像格式支持 YUV (422/420)、YCbCr422、RGB565 以及 JPEG 格式, 根据不同的分辨率设置, 摄像头输出帧率从 15-60 帧可调。当系统工作时, 首先由寄存器配置模块配置摄像头内部寄存器参数来设置工作模式, 寄存器配置完成后, 摄像头会输出指定频率的时钟, 像素数据、行、帧同步信号也会将摄像头的时钟同步输出。这些信号被 FPGA 接收后经过数据拼接、白平衡算法处理、SDRAM 缓存并跨时钟域传输并处理后按照 VGA 时序进行传输到 LCD 显示屏上。观察视频流处理的最终效果。

## 二、系统设计

### (一) 摄像头驱动

摄像头由上电到正常工作需要经过上电和寄存器配置以及摄像头输出数据的处理三个过程, 其中寄存器配置采用的是 SCCB 协议。其中上电只需要将摄像头 RESET 和 PWDN 两个管脚置指定电平一段时间即可, 这里不做过多介绍。寄存器配置可细分为两个模块: 一个模块是 SCCB 协议模块, 该部分按照 SCCB 协议向外发送数据; 另一个是 SCCB 控制器, 控制器放着需要向寄存器写入的数据和控制着 SCCB 协议模块的数据发送。在这里我们设置摄像头的工作模式为输出像素格式 RGB565 和有效像素阵列 1024 × 600 以及输出帧数三十帧。

摄像头正常工作后会向外发送像素信息和行帧同步信号, 但是像素信息需要经过一定的处理才能变为 RGB565 格式的图像数据。摄像头的的数据引脚只有八个, 所以一个时钟周期只能发送八位数据, RGB565 格式下一个像素点为 16 位数据, 查询 OV5640 的文档可知摄像头将一个像素点的数据分两个时钟周期发送, 先发数据的高八位, 再发数据的低八位, 所以我们可以以此信息来构建图像采集和处理部分。

在行同步信号的作用下用数据高低标志来标志当前时钟周期的数据, 如果当前是高位八位像素信息, 则 ov5640 输出的像素值赋值给输出像素的高八位; 如果当前标志为低, 则将该值赋值给低八位。

### (二) 白平衡算法模块

本实验加速的算法为灰度世界算法。世界算法是以灰度世界假设作为基础的, 该假设认为对于一副有着大量色彩变化的图像, R、G、B 三个分量的平均值趋于同一个灰度  $K$ [7]。因此该算法的第一步就是求出灰度值  $K$ 。计算灰度值一般有两种: 一种是计算时直接将灰度值设置为定值 (一般为 128), 另一种计算方式是计算当前图像的灰度值。计算当前图像的灰度值方法如下:

先计算各个通道的像素和。其中 row 和 col 分别表示一副图像的行总数和列总数。

$$R_{sum} = \sum_{i=1}^{row} \sum_{j=1}^{col} R$$

$$G_{sum} = \sum_{i=1}^{row} \sum_{j=1}^{col} G$$

$$B_{sum} = \sum_{i=1}^{row} \sum_{j=1}^{col} B$$

然后再计算各个通道的均值  $R_{avg}$ 、 $G_{avg}$  和  $B_{avg}$

以及图像的灰度值  $K_{avg}$ 。

$$R_{avg} = \frac{R_{sum}}{row \times col}$$

$$G_{avg} = \frac{G_{sum}}{row \times col}$$

$$B_{avg} = \frac{B_{sum}}{row \times col}$$

$$k_{avg} = \frac{R_{sum} + G_{sum} + B_{sum}}{3 \times row \times col}$$

第二步是计算各个通道的增益  $K_r$ 、 $K_g$ 、和  $K_b$

$$K_r = \frac{K}{R_{avg}} \quad K_g = \frac{K}{G_{avg}} \quad K_b = \frac{K}{B_{avg}}$$

第三步根据 Von Kries 对角模型, 用原图像的各个通道的分量值乘以对应通道的增益值得到新图像的像素值, 此过程中过大且图像增益大于 1 时根据公式可知

像素值可能出现溢出 (像素值大于 255) 的现象, 这里我们将溢出值直接置为 255。

$$\begin{bmatrix} R_{new} \\ G_{new} \\ B_{new} \end{bmatrix} = \begin{bmatrix} K_r & 0 & 0 \\ 0 & K_g & 0 \\ 0 & 0 & K_b \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

此过程中过大且图像增益大于 1 时根据公式可知像素值可能出现溢出 (像素值大于 255) 的现象, 这里我们将溢出值直接置为 255。

### (三) 白平衡算法的 FPGA 实现

由上述公式可知, 求解一副图像的白平衡算法需要将图像遍历两遍, 但是在 FPGA 中, 图像是以数据流的形式传输的, 如果遍历两遍将要耗费大量的存储资源和逻辑资源以及系统延时。考虑到在一个三十帧的视频流中帧与帧之间的像素数据差距一般很小, 且在实际应用中如果采用更高帧率的相机, 帧与帧之间的差距还会进一步缩小, 因此我们选择的方案是利用上一帧图像中的各个通道的均值和通道增益, 来计算下一副图像的像素值, 这样做损失了少量的白平衡算法效果但是使得白平衡算法在 FPGA 内部的视频流延时大大降低并且也降低了系统的复杂度[8]。

摄像头采集的数据经过处理后输入白平衡算法模块, 当一帧图像输入时, 白平衡算法模块将像素数据分两路传输: 一路执行白平衡算法的第一步先将各个通道的值累加并且等待一帧传输结束后更新通道均值和灰度值, 另一路数据利用上一帧计算出的通道均值和灰度值计算出输出的像素值。

计算通道增益时需要保留浮点数进行后续计算, 但是由于 FPGA 内部都是数字电路, 在量化浮点数时会消耗较多的逻辑资源并且也会给后续的浮点数计算带来困难。因此我们将第二步计算通道增益和第三步计算新的像素值结合起来得到新公式

$$R_{new} = \frac{R \times k}{R_{avg}}$$

$$G_{new} = \frac{G \times k}{G_{avg}}$$

$$B_{new} = \frac{B \times k}{B_{avg}}$$

然后先调用乘法后调用除法 IP 核, 避免了浮点数计算从而提高计算准确度。此算法模块利用了 FPGA 的并行思想和流水线思想, 并针对图像数据特点和 FPGA 硬件结构进行了算法的实现优化, 将白平衡算法执行速

度提升至数十个时钟周期。相比于 CPU 的串行计算和两次遍历，提升速度极大。

#### (四) 图像缓存模块

图像缓存是视频流里面一个非常重要的模块，它提供了最少能容纳一副完整图像的内存，来保证读取图片时能取到完整的图像像素信息。本次显示的图像的大小为  $\frac{1024 \times 600 \times 16}{8 \times 1024 \times 1024} \approx 1.17M$ ，FPGA 内部的 ram 内存过小，不满足缓存需要，并且当作为视频源的缓冲内存时，为了防止读写图像数据的速率不一致而造成图像撕裂的情况，我们需要将内存划分成两个区域来存储两帧图像供读写进行乒乓操作，因此我们外接了一个 256Mbit 内存的 SDRAM 来进行缓存。

SDRAM 为同步动态随机存取内存，全称为 synchronous dynamic random-access memory，本次采用的 SDRAM 型号为第一代 SDR SDRAM，为单沿 SDRAM，型号为 W9825G6KH，具有 13 位地址尾款，16 位数据位宽。单沿只在时钟上升沿进行数据采样和数据发送，

#### (五) VGA 驱动模块

SDRAM 内部的数据需要遵循 VGA 的时序要求发送给 LCD 显示器才能被正确的接收，VGA 显示屏从左到右、从上到下依次扫描。在行扫描周期内有同步、后沿、有效像素、前沿四个部分；场扫描周期也有同步、后沿、有效像素、前沿四个部分。当扫描到同步阶段时，将同步信号拉低，当扫描到数据有效区时发送一个像素数据，在 60 帧显示下每秒钟扫描 60 次显示区域。根据 VESA 协议，VAG 的驱动时钟应该为 50M。

#### 三、实验结果及分析

对仿真模块编写仿真文件，输入一副  $3 \times 3$  大小的图像计算图像的各个通道的均值和图像的灰度值。经计算仿真无误。

对于路径二，输入下一个  $3 \times 3$  图像矩阵，利用上一个矩阵算出的通道均值和灰度值计算当前图像的均值和灰度值。经计算仿真无误。

对于系统延迟分析，路径一计算上一幅图像的通道均值和灰度值经过了一个除法器，按照除法器延迟最大的路径灰度值  $k_{avg}$  来算，除法器耗时 33 个时钟周期，在  $1024 \times 600$  和六十帧的分辨率下，像素时钟约为 50 MHz，所以路径延迟约为 660ns。路径二经过原始数据经过一个乘法器一个除法器，乘法器固定耗时 3 个时钟周期，除法器耗时 18 个时钟周期，再加上为了数据稳定数据经过了一个寄存器延迟了一个时钟周期，共耗时 22 个时钟周期 440ns。由于硬件结构和算法的设计，在

实际数据传输的过程中仅需要考虑数据延迟，因此采用路径二的延迟 440ns。

在实际处理效果上，处理前的图片偏色很严重，图像色彩略单调，而经过白平衡处理后的图像在色彩上更加贴近原色，与人眼看到的图像更为贴近。

由仿真和实际处理可以看出，经过 FPGA 加速的白平衡算法处理速度快、算法结果精确。四、结束语

本文对传统的自动白平衡算法进行了硬件实现，运用 FPGA 并行思想和流水线思想，将算法移植到硬件电路中并针对算法进行了硬件优化，在不影响算法性能的前提下大幅提高了算法的计算速度以满足图像处理的实时性。虽然本次实验以  $1024 \times 600$  分辨率 50M 时钟下进行仿真并实验得出的延迟约为 440ns，但是由于数字电路的优越性，在一定范围内，随着图像分辨率的增加而时钟频率变快但处理周期不变，系统的数据延迟还会进一步降低。具有较高的应用价值。

课题来源：研究生校级创新项目；课题名称：基于 FPGA 的神经网络设计与实现；课题编号 CX2022SY043。

#### 参考文献：

- [1]王红,彭亮,于宗光.FPGA 现状与发展趋势[J].电子与封装,2007(07):32-37.
- [2]王红,彭亮,于宗光.FPGA 现状与发展趋势[J].电子与封装,2007(07):32-37.
- [3]杨超,杜正聪,印茂伟,董加强,芮志聪,须永朋,赫培峰.基于 FPGA 的疲劳驾驶检测系统设计[J].实验科学与技术,2022,20(05):1-6.
- [4]陈鑫元,李筠,杨海马,宋佳,邵璐先.自适应阈值图像二值化及形态学处理的 FPGA 实现[J].电子测量技术,2016,39(07):67-71.
- [5]张崇关,刘城,惠肇宇.基于 FPGA 的 SAR 图像压缩实现[J].工业控制计算机,2022,35(10):50-52.
- [6]张棋,贾方秀,殷婷婷.基于改进 Sobel 算法的实时图像边缘检测系统设计[J].仪表技术与传感器,2018(02):101-104+108.
- [7]贾晨鑫,张荣福,任梦圆,徐朝辉,龚姬央丹,刘翰林.基于 FPGA 图像处理的内窥镜焊点检测系统[J].光学仪器,2022,44(05):53-60.
- [8]张晓飞,袁祥辉.基于 DSP 成像系统的视频图像采集部分的实现[J].压电与声光,2002(03):247-250.
- [9]张韫皓,黎震海,谢泽鸿,覃壬镜.基于 FPGA 的实时图像去雾算法改进[J].长江信息通信,2022,35(10):66-68.