

基于边缘点的动态 ROI 图像传输系统与 FPGA 实现

陈 南 安 然 张立平 郝汝栋
(易安基自动化设备(北京)有限公司 北京 100176)

摘 要: 主要阐述了一种运用在工业检测中的图像采集、裁剪技术。首先,指出现有工业光学检测中遇到的传输瓶颈问题,并讲解基于边缘点的动态 ROI 裁剪的算法原理,通过实时 Canny 边缘检测定位特征点,生成一定像素宽度的纵向 ROI 窗口实现数据压缩。接着提出了对 GVSP 协议的修改,包括新增 Index 包并修改 Leader 包,以适配动态 ROI 的图像数据传输。随后详细说明如何在 FPGA 平台上实现动态 ROI 算法,采用 BRAM 替代 1 280 位宽寄存器优化存储资源,结合 128 位总线设计消除内部带宽瓶颈。最后,通过实验测试以及工业运用场景开展验证,结果表明该系统在保持原始灰度精度的前提下显著降低了有效数据量,可支持多相机数据传输。

关键词: 板带;测宽;孔洞;FPGA;图像压缩;网络传输

中图分类号: TN23 **文献标识码:** A **国家标准学科分类代码:** 510.4050

Edge-driven dynamic region-of-interest transmission for industrial vision systems on FPGA

Chen Nan An Ran Zhang Liping Hao Rudong
(Yianji Automation Equipment (Beijing) Co., Ltd., Beijing 100176, China)

Abstract: This paper presents an image acquisition and cropping technique for industrial inspection systems. Initially, it identifies the transmission bottleneck in existing industrial optical detection and elaborates on the algorithmic principle of dynamic ROI cropping based on edge points. By performing real-time Canny edge detection to localize feature points, vertical ROI windows with configurable pixel widths are generated to achieve data compression. Subsequently, modifications to the GVSP are proposed, including the addition of Index packets and adaptation of Leader packets to accommodate dynamic ROI-based image data transmission. The implementation of the dynamic ROI algorithm on an FPGA platform is then detailed, where BRAM substitution for 1 280-bit registers optimizes storage resources, and a 128-bit bus design eliminates internal bandwidth bottlenecks. Experimental verification and industrial deployment demonstrate that the system significantly reduces effective data volume while preserving raw grayscale precision, enabling reliable multi-camera data transmission.

Keywords: steel strip; width measurement; hole defect; FPGA; image compression; network transmission

0 引 言

板带材作为钢铁、有色金属工业的核心产品,其几何尺寸精度与表面质量直接影响下游制造业的成品率与安全性。实时高精度板带宽度测量、缺陷检测不仅是质量控制的基石,更是保障生产安全、降低经济损失的核心技术环节。

使用相机阵列进行视觉检测是一种常见的板带检测手段^[1]。通常辅以背光光源照射,图像对比度高、边缘锐利,对微小孔洞缺陷的检出很有帮助。

而在视觉检测中,检测精度和板带速度之间存在着不

可兼得的矛盾。检测精度要求更小的像素尺寸、更多的相机,板带速度则要求更高的采样频率;这两者都会增加图像采集的数据量,在图像数据传输、处理速度一定的情况下,往往只能顾此失彼,限制了图像检测系统的性能提升。

在背光照明模式下,图像对比度较高。相机采集的图像中仅边缘与缺陷(如孔洞、边裂)区域蕴含有效信息,此外

的超过 95% 的像素为冗余黑白像素。

而在图像数据传输方面,传统 GigE Vision 流协议(GigE Vision streaming protocol, GVSP)^[2]强制传输全幅图像,使得以太网成为系统瓶颈。如,当单个相机的分辨率达 1 280×960 pixels 时,单帧就有 9.8 Mb 数据;高精度检

测常常需要几十、甚至上百个相机组成的阵列,即使是万兆网,帧率也将被限制在个位数,难以满足高速产线需求。

现有改进方案往往陷入两难境地:跳行采样虽可提升传输效率,但会损失精度;基于软件的亚像素算法虽能提高精度^[3],却需原始图像数据支撑,不允许对图像进行压缩^[4-5]。

为突破这一困局,本文提出基于边缘点的动态感兴趣区(region of interest, ROI)裁剪机制、以及 GVSP 协议扩展架构。该方案依托现场可编程门阵列(field programmable gate array, FPGA)的流水线处理能力,在图像采集同时完成 Canny 边缘检测,并基于边缘点坐标动态生成一定宽度的纵向 ROI 区域,仅传输包含有效特征的图像片段。通过重构 GVSP 协议包结构与优化片内存储架构,在保持原始像素值不变的前提下实现高比例带宽压缩,为高速、高精度工业检测开辟新路径。

近年来基于 FPGA 图像处理的研究相当广泛^[6-8]。利用 FPGA 的高速处理能力,进行图像采集存储^[9];在 FPGA 平台上实现深度学习算法^[10],能够直接识别复杂的图像中的物体、缺陷;用 FPGA 实现插值法增加图像分辨率^[11];用 FPGA 实现深度神经网络图像压缩^[12]等。

但本文提出的算法更节省资源,其运算不涉及浮点数,使得一片 CycloneV 芯片即可支持 10 个图像传感器的采集、预处理和传输。

此算法还更节省时间,不需要在接收完整图像后再消耗时间进行处理,而是接收一个像素处理一个像素,不会影响原有的图像采集速度。

当然,相比上述复杂算法,本算法的精度有所欠缺、更没有智能识别的能力;但本算法只是图像采集的一个环节,用于粗略识别“有效图像”,将其裁剪出来传输给个人计算机(personal computer, PC);检测需要的精度自会由 PC 的精密算法来实现。

1 算法原理

1.1 背光检测图像的特点

背光照明、单色相机采集,获得的图像对比度通常较大,被板带遮挡的部分为黑色(像素值接近 0),板带之外或孔洞等透光的部分为白色(像素值接近 255)。如图 1 所示。

可见,画面中大部分都是无意义的黑、白像素,仅板带边缘、孔洞附近的图像有意义。

1.2 FPGA 图像处理的特点

FPGA 因其并行结构,导致在 FPGA 上实现的图像处理算法和 CPU 上实现的截然不同。

CPU 的图形算法,常需要将图像数据都导入 RAM 中再进行操作。是完全接收图像之后,再花时间进行处理。

而 FPGA 的处理是实时的,接收一个像素就处理一个像素。图像接收完毕,也就处理完毕,处理速度永远与接收速度一致,只有流水线延时。

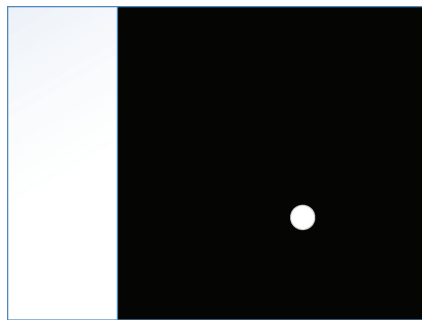


图 1 背光检测图像示意图

Fig. 1 Backlighting detection schematic

这就使得 FPGA 可以在控制相机芯片(甚至是多个相机芯片)进行图像采集的同时,还可以同时进行图像处理,如 Canny 边缘检测^[13],正因如此,本文的算法才有可行性。

1.3 边缘检测算法

对于相机采集的图像(图 1),首先要提取出边缘点,也就是黑白交界的点。这里采用了 Canny 算法^[14],大致流程如下。

1) 首先用高斯滤波去除特异像素点的影响。

2) 对滤波后的图像套用 Sobel 算子^[15]。

3) 对 Sobel 之后的粗略边缘图进行非极大值抑制,使边缘线“变窄”。

4) 对抑制之后的图像进行 Canny 阈值筛选。

5) 最终得到边缘图,边缘点为 255,其余像素都为 0。

如图 2 所示。

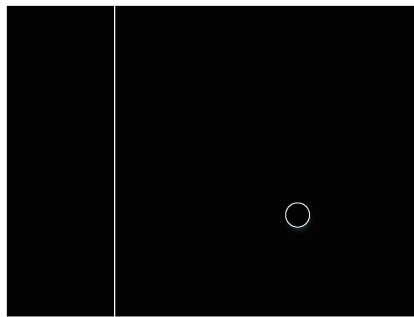


图 2 经过 Canny 算法得到的边缘图

Fig. 2 Canny edge detection result

1.4 ROI 提取和套用

ROI 即图像中的“有效”部分。为了获取 ROI 的范围,对图 2 中的每一个边缘点,左右扩展 5 个像素,形成上下通长的 ROI 区域。如图 3 所示。

这 5 个像素的宽度是可调的,根据图像边缘特性决定;如果图像边缘模糊、绵延较广,可以设大一些。

这里采用上下通长的 ROI 形状,是为了贴合应用场景以及简化处理流程。板带检测中每一帧都会有竖直方向的边缘,所以采用纵向的长条可以最小化 ROI 的面积^[16]。而长条 ROI 相比小方块 ROI,只需记录一个维度的信息,处

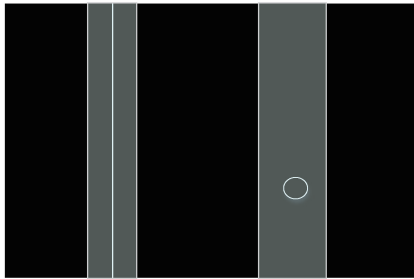


图 3 提取 ROI 示意图

Fig. 3 ROI extraction

理起来比较方便。

下一步是将 ROI 套用到原图上。如图 4 所示。



图 4 将 ROI 套用到原图

Fig. 4 Mapping ROIs onto the original image

将 ROI 之外的原图去除,再将几个 ROI 合并,如图 5 所示。

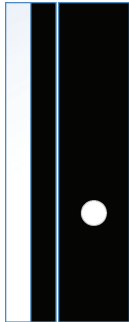


图 5 仅保留 ROI 内数据

Fig. 5 Retaining only data within ROIs

由于 ROI 是上下通长的长条,所以合并后,图像的高度没有变化,而宽度大大降低,达到了数据压缩的目的。

1.5 算法对图像精度的影响

在相关应用场景中,FPGA 只是负责驱动相机芯片进行图像采集,并将图像发送给 PC,由 PC 进行精细的图像处理。

因此,PC 的算法才影响检测精度,而 FPGA 只需保证图像质量不损失,就不会影响检测精度。

本文提出的动态 ROI 裁剪算法,只是在 FPGA 端去除了图像中的无效数据,而有效部分完全保留了原始像素,因

此采用本文算法不会影响系统的检测精度。

相反,减少了图像的数据量,会降低 PC 端算法的压力,可以使整体系统的检测效率进一步提升。

2 协议修改

《GigE Vision》规范是自动成像协会 (Automated Imaging Association, AIA) 发布的协议规范,解决工业相机与主机间高速数据传输和设备控制的标准化问题,通过以太网实现千兆级别的图像传输。规范中的 GVCP 协议用于设备控制,GVSP 协议用于图像数据传输;二者都是建立在 IP/UDP 协议栈之上的应用层协议。

根据原版 GVSP 协议,在图像传输过程中,每帧的长宽大小必须一致。

每帧图像由一个 Leader 包、多个 Payload 包、和一个 Trailor 包组成。其中,有效的图像数据都在 Payload 包中。在开启图像传输前,必须约定好 Payload 包的长度和一帧数据的总长度。即 Payload 包的数量也被固定下来。

而在本研究修改后的协议中,每帧图像是由数个 ROI 拼接而成;每帧的 ROI 数量、大小均不相同,甚至有可能完全没有 ROI。

因此需要对 GVSP 协议进行一定的调整。

首先在 Leader 包中增加本帧的图像长宽信息,即图像数据总长度。

其次,当一帧图像完全没有 ROI 时,即没有图像数据传输;为了在这种情况下让主机掌握采集状态、校正同步信息,增加一种新的数据包:Index 数据包。对每一帧图像而言,Index 数据包用于向主机通告,本帧每个相机是否有 ROI;此外还包含帧 ID 等信息。一台设备的每次触发必定会产生一个 Index 包;而只有设备的某些相机包含 ROI,后续才会跟随发送 Leader 包、Payload 包、Trailor 包组成的图像数据。

2.1 Index 包格式

Index 包格式如表 1 所示。前 8 byte 采取和原版协议的 Leader 包等相同的包头格式。其中 format 字段定为 0。

表 1 Index 包格式

Table 1 Index packet format

字节序	内容	取值
0~1	status	0
2~3	block id	—
4	format	0
5~7	packet id	0
8~11	size_x	—
12~15	size_y	—
16~19	offset_x	—
20~23	offset_y	—
24~33	相机状态表	—
34~35	保留	0
36~39	时间戳	—

在同一帧图像中,所有的 Index 包、Leader 包、Payload 包、Trailer 包的 block id 是同一个值。并且,连续的图像帧的 block id 也应连续。主机软件可以根据这一字段判断丢帧情况。

Index 包的 packet id 固定为 0。之后同一帧数据内所有类型数据包的 packet id 连续增长。

Index 包的 size_x、size_y、offset_x、和 offset_y 字段表示的是原图(图 1)的尺寸和相机采集时设置的偏移量。

相机状态表字段有 10 byte,代表 10 个相机,因为 FPGA 的并行属性,1 片 FPGA 可以控制多达 10 个相机。字段取值如下。

0:代表对应相机本帧采集正常,而且图像不包含 ROI。

0x80:代表对应相机本帧采集正常,而且图像包含 ROI。

0xff:代表对应相机本帧采集不正常。

时间戳字段是 4 byte 整数(单位为:μs)。

2.2 Leader 包格式

Leader 包格式基本与 GVSP 原版相同,如表 2 所示。

表 2 Leader 包格式
Table 2 Leader packet format

字节序	内容	取值
0~1	status	0
2~3	block id	—
4	format	1
5~7	packet id	—
8	相机号	0~9
9	reserved	0
10~11	payload type	1
12~15	保留	0
16~19	时间戳	0
20~23	pixel format	0x01080001
24~27	size_x	—
28~31	size_y	—
32~35	offset_x	—
36~39	offset_y	—
40~43	padding	—

修改的地方有:

占用了原本是保留字节的第 9 byte,用于表示这个图像帧属于设备的第几个相机,取值范围 0~9。

size_x 和 size_y 字段的格式没有改变,但是意义有所改变。现在这两个字段代表的是有效窗口拼接完成的图片的宽和高(图 5),而非原始图片。这两个字段的乘积就是本帧图像的数据长度。

时间戳字段是 4 byte 整数(单位为:μs)。

2.3 参数行格式

如图 5 所示,1 帧图像可能是由几个 ROI 拼接而成的,

每个 ROI 的宽度不低于 6 pixel。为了传递每个 ROI 的位置信息,将最后一行作为参数行。其格式定义如表 3 所示。

表 3 参数行格式
Table 3 Parameter format

字节序	内容	取值
0	起始标志	255
1	相机序号	0~9
2~3	窗口起始 x 坐标	≤1 279
4~5	窗口宽度	≥11

2.4 兼容性分析和 PC 软件协议适配

支持修改后协议的设备发送的图像数据对于支持标准 GigE Vision 的软件来说是无法解析的,会显示混乱。因此,PC 软件必须修改或重做 GVSP 的协议包,来适配修改后的协议。

1)每帧图像必定以一个 Index 包作为起始,PC 软件应该从中获取:完整图像尺寸,用于显示图像;block id,用于验证是否存在丢帧情况;相机状态表,用于获取采集状态,以确定哪些相机在本帧生成了有效图像。

2)Index 包之后,设备将会以相机为单位依次传图。每个相机的 ROI 图像会封装为“Leader 包-若干个 Payload 包-Trailor 包”的形式,这点和原版 GVSP 协议一致。PC 软件要从 Leader 包中获取 size_x 字段,代表 ROI 图像的宽度,这很重要,因为每个相机、每一帧的 ROI 图像的宽度都可能不同。

3)每一个 ROI 图像的最下一行是参数行,PC 软件需要从中读出“窗口起始坐标”和“窗口宽度”以确定 ROI 图像在整图中的位置。

3 FPGA 实现

3.1 FPGA 实时高斯滤波、Canny 的实现

高斯滤波^[17]、Canny 的算子套用,都需要以“卷积窗口”为单位对图像进行处理,即当前像素、与其周边的 8 个像素所组成的 3×3 pixels 图像区域。

在 FPGA 中,首先使用两个移位器获取两行像素的缓冲,则移位器的输出和当前像素就构成了图像中同一列的 3 个相邻像素。如图 6 所示,扫图过程中,点(3,2)是新像素,之前的像素压入两级移位器中,则两个移位器的输出正好是点(3,0)和点(3,1),与新像素点(3,2)构成一列的 3 个相邻像素。

此时将这个 1×3 的列缓冲两个周期,就得到了 3×3 的卷积窗口。如图 7 所示,此即为以点(2,1)为中心的 3×3 卷积窗口,用于进行高斯滤波、Canny 等算法的实时操作。

由此,每多一个实时图像处理步骤,输出的图像就会有 1 行的延时。但处理速度和接收速度是一致的,不会造成数据堆积。

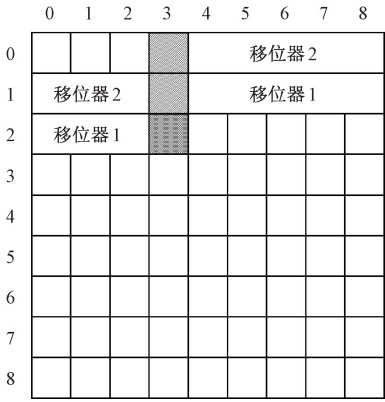


图 6 两级移位器获得同列 3 像素

Fig. 6 Acquisition of three vertically adjacent pixels via two-stage shifter

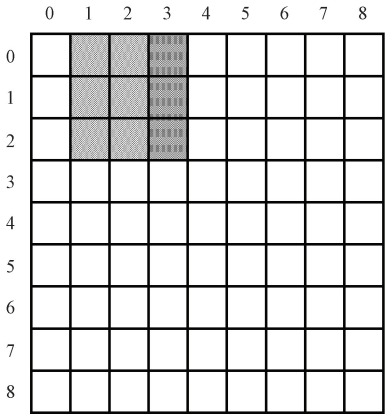


图 7 缓冲两个周期获得 3×3 卷积窗口

Fig. 7 Constructing 3×3 convolution window via two-cycle buffering

至于具体高斯滤波和 Canny 的实现方式,已经有很多研究^[18-21],不在这里赘述。

3.2 ROI 的标注

对于 Canny 模块输出的边缘图再次进行扫描。因为本设计的 ROI 是条形的,所以 ROI 只需要记录 X 坐标。

建立一个寄存器来存储 ROI。每一个位对应图像的一列;如果该位置 1,则表示对应列在 ROI 中。

这个寄存器的宽度会和图像宽度相同。这么大的寄存器会非常占用资源,因此将它优化为 RAM,在宽度和深度之间找到平衡。但坏处是就不能实时读写了,好在扫描图像时,X 坐标是递增的,完全可以预估,所以可以提前进行预读。

3.3 图像数据的转存和组帧

图像输入 FPGA 后,分成两支,如图 8 所示。

第 1 支图像数据先后通过高斯滤波模块、Canny 模块、ROI 提取模块,得到存有 ROI 信息的 RAM。

第 2 支图像数据通过高斯滤波后,通过 DMA 存入片外 DDR RAM,是完整的原图。当全图接收完毕时(还要留出一定延时以供第 1 支的 ROI 提取完成),再通过另一个

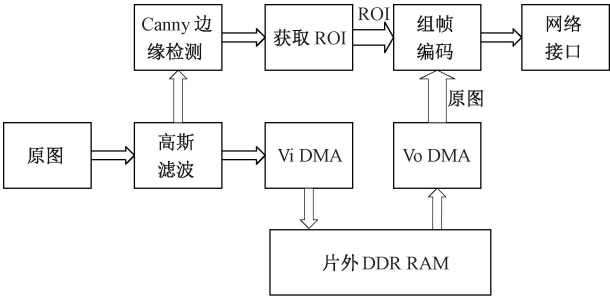


图 8 图像数据处理数据流

Fig. 8 Image processing dataflow

DMA 读出,进入组帧模块。

组帧模块会根据 ROI 信息对图像信息进行筛选。如果该像素位于 ROI 中,就进行打包;如果该像素位于 ROI 之外,则丢弃。这样就获得了拼接后的 ROI 内的图像。

最后组帧模块再把图像数据按照 GVSP、UDP、IP、Ethernet 协议顺序进行层层打包,计算校验和等,转给网络接口。

3.4 数据带宽设计

在调试中,本研究发现,一个 FPGA 控制的 10 个相机中,只有前 3 个相机的数据能发送出来,之后的 7 个相机都没有数据。怀疑是采样周期内剩余的时间不够发送下一个相机图像;经过一系列调试发现确实如此,而且单个相机图像传输的时间远长于预期,约为预期的 2 倍。

为了分析解决这个问题,把组帧模块前后的数据带宽标注出来。

如图 9(a)所示,从 Vo DMA 开始到以太网模块为止,所有的总线带宽均为 32 b×50 MHz,即 1 600 Mbps,可以覆盖以太网 1 000 Mbps 的带宽,内部总线不成为瓶颈。

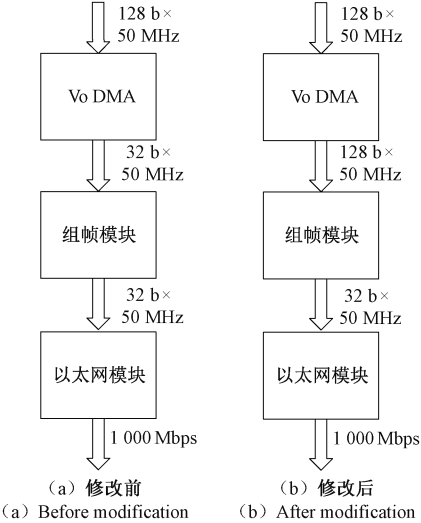


图 9 修改前后的带宽设计

Fig. 9 Bandwidth architecture: Pre-vs post-modification

但这是在全图传输的情况下。如果是动态 ROI 传输,组帧模块对数据进行了大比例的裁剪,导致输出数据量远

低于输入数据量。这样一来,组帧模块的输入端就成为了瓶颈,若不扩充带宽,传输速度就难以提升。

为了验证带宽扩充的可能性,需要一级一级调查数据链路上游的带宽。先是 Vo DMA 模块的数据输入端,是 $128\text{ b} \times 50\text{ MHz}$ 的 Avalon-MM 总线;再之前是 DDR 控制模块接口的 $128\text{ b} \times 125\text{ MHz}$ 的 Avalon-MM 总线;再之前是 DDR 芯片读写接口 $32\text{ b} \times 800\text{ MHz}$ 。

核算下来数据链路上游的瓶颈是 $128\text{ b} \times 50\text{ MHz}$,即 $6\,400\text{ Mbps}$ 。也就是说,如果能把组帧模块的输入端带宽也扩充到 $128\text{ b} \times 50\text{ MHz}$,就是能在设计上做到的极限。而按照之前的比例,等于是扩充到 4 倍;之前足够 3 个相机图像的传输,扩充后应该足够 12 个相机数据传输,也满足本研究 10 个相机的需求,如图 9 右侧所示。

4 验证测试

4.1 软硬件平台

开发环境是 Quartus Prime 18.0.0。FPGA 芯片选用了 Cyclone V 5CEFA9,面阵相机采用了 AR0135,一片 FPGA 可以操控 10 个相机芯片,并驱动一个千兆以太网接口用于图像数据导出。

FPGA 内建 CPU,运行 C 程序建立网络协议栈,并控制相机;Verilog 逻辑编码用于处理图像的采集、处理、转存、组帧、发送。逻辑部分的时钟为 $50 \sim 100\text{ MHz}$,相机芯片输出的像素频率为 72 MHz 。

4.2 资源占用

在已有的一个 FPGA 工程项目上,添加本研究的模块。原工程项目已经实现功能:用 Cyclone V 5CEFA9 采集 10 个相机芯片的图像信息,打包成为 GVSP 协议发送到 PC。原工程项目的逻辑资源占用率是 40%。

在实现 1.4 节提到的“ROI 提取”时,原本的设计是构建一个 $1\,280$ 位(等同于画面行宽)的寄存器,用于标记哪些列处于 ROI 中。功能实现正常,但这种设计及其消耗资源,设计还未完全实现时,占用率就高达 90%。

因此改变本研究思路,将 $1\,280$ 位宽寄存器改为 $32(\text{宽}) \times 40(\text{深})$ 的片内双口 RAM。优势是确实省资源了,完成整个设计后,占用率只有 50%;代价是读操作要消耗 2 个时钟周期,无法做到实时读写。

好在图像是按顺序每行从左到右扫描的,即不会有随机地址读写的情况,这样就可以提前两周期读出数据存在临时寄存器中,规避了这个问题,成功实现了设计。

4.3 实验室测试

被测转盘线速度 200 m/min ,相机曝光时间 $200\text{ }\mu\text{s}$ 。测试系统共有 6 台设备,总共 60 个相机,相机分辨率为 $1\,280 \times 960\text{ pixels}$ 。在相同帧率下,分别采用全图传输和本文所述的动态 ROI 裁剪模式进行测试,比较二者生成的图像数据量。

如图 10 所示,全图传输时,图像分辨率 $1\,280 \times$

960 pixels ,当帧率达到 8.8 时,可见图像数据量已接近千兆网的上限 949 Mbps ,即帧率没有了提升的空间。



图 10 全图传输的数据量

Fig. 10 Data volume of full picture transmission

运用本文所述的动态 ROI 裁剪模式时,图像分辨率仍然设为 $1\,280 \times 960\text{ pixels}$,因为裁剪掉了无效数据,可见当帧率同样为 8.8 时(图 11),图像数据量仅为 6.6 Mbps ,压缩比达到 143 倍。帧率还有巨大的提升空间。

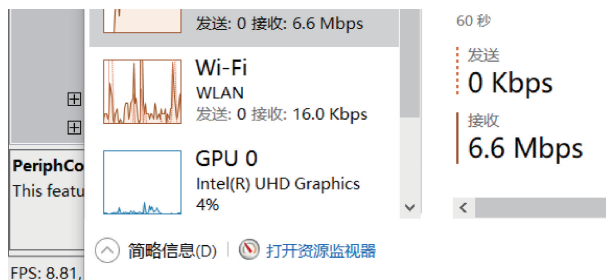


图 11 动态 ROI 图像传输的数据量

Fig. 11 Data volume of dynamic ROI transmission

4.4 应用测试

本文所述技术,已经应用在实际产品中,服务于多家钢铁企业的带钢产线。在保证 0.2 mm 的检测精度的同时,满足 500 m/min 的产线速度要求。

5 结 论

本研究实现了一种基于边缘检测的动态 ROI 裁剪算法,并通过对 GVSP 协议的扩展,构筑了一种新型的工业视觉传输系统,能够在不影响采集精度、不修改像素信息的前提下,通过删减无效图像数据,将传输数据量缩减为之前的 1%级别。同时,因为采用的是裁剪,而非压缩的方式,减轻了数据接收端的解码压力。为可能的亚像素处理创造了条件。

经过实验室测试,性能达到设计目标。为高速、高精度工业检测提供了有效的图像传输手段,具有较强的实用价值。

参考文献

- [1] 张晓春. CCD 测宽仪在热轧板带中的应用[J]. 宝钢技术, 2015(4): 73-77, 80.
ZHANG X CH. Application of CCD width gauge in hot rolling strip[J]. Bao-Steel Technology, 2015(4):

- 73-77,80.
- [2] 吕鹏. GigE Vision 协议数据流控机制的分析与实现[J]. 电子世界, 2019(19):113-115.
LYU P. Analysis and implementation of data flow control mechanisms in the GigE Vision protocol[J]. Electronics World, 2019(19):113-115.
- [3] 曾猛杰,汪晨曦,赖俊杰,等. 亚像素边缘检测算法综述[J]. 光学精密工程, 2024,32(23):3513-3524.
ZENG M J, WANG CH X, LAI J J, et al. A Review of sub-pixel edge detection algorithms[J]. Optics and Precision Engineering, 2024,32(23): 3513-3524.
- [4] 周焱江,李锦明. 基于多角度的 JPEG 算法优化的硬件实现[J]. 舰船电子工程, 2024,44(12):126-131,214.
ZHOU Y J, LI J M. Hardware implementation of JPEG algorithm optimized from multiple perspectives[J]. Ship Electronic Engineering, 2024,44(12):126-131,214.
- [5] 王成昊. 基于 FPGA 的低复杂度 JPEG 解码系统设计及实现[D]. 成都:四川师范大学, 2024.
WANG CH H. Design and implementation of low complexity JPEG decoding system based on FPGA[D]. Chengdu:Sichuan Normal University, 2024.
- [6] 聂晨,余江南,潘晓琴. 基于 FPGA 的信号调制方式识别和参数估计设计[J]. 现代信息科技, 2024,8(13):19-23,28.
NIE CH, YU J N, PAN X Q. Design of signal modulation method identification and parameter estimation based on FPGA[J]. Modern Information Technology, 2024,8(13):19-23,28.
- [7] 马骁. 基于 SoC FPGA 的高动态图像处理系统研究[D]. 杭州:浙江大学, 2021.
MA X. Research on high dynamic image processing system based on SoC FPGA[D]. Hangzhou:Zhejiang University, 2021.
- [8] 马浚轩,李红,张旭,等. 面向光轴指向测量系统的光斑质心高精度实时解算方法[J]. 仪器仪表学报, 2024,45(6):266-273.
MA J X, LI H, ZHANG X, et al. High-precision real-time algorithm for spot centroid determination in optical axis pointing measurement system[J]. Chinese Journal of Scientific Instrument, 2024,45(6):266-273.
- [9] 王振,黄春平,陈威威. 基于 FPGA 的 SATA 图像采集系统的设计与实现[J]. 电子测量技术, 2025,48(14):26-34.
WANG ZH, HUANG CH P, CHEN W W. Design and implementation of a SATA image storage system based on FPGA[J]. Electronic Measurement Technology, 2025,48(14):26-34.
- [10] 安瑞,冉梅梅,胡艳艳. 基于深度学习的纸病检测系统开发研究[J]. 造纸科学与技术, 2025,44(10):127-130.
AN R, RAN M M, HU Y Y. Research on the development of paper defect detection system based on deep learning[J]. Paper Science and Technology, 2025,44(10):127-130.
- [11] 丁士强,梁玉峰,尚嵩. 视频图像边缘自适应插值算法及其 FPGA 实现[J]. 自动化应用, 2025,66(16):115-117.
DING SH Q, LIANG Y F, SHANG S. Edge-adaptive interpolation algorithm for video images and its FPGA implementation[J]. Automation Application, 2025,66(16):115-117.
- [12] 宋赛,崔昭,詹尹僧,等. 面向深度神经网络图像压缩的高性能算术编码硬件设计[J]. 电子与信息学报, 2025,47(9):3230-3240.
SONG S, CUI ZH, ZHAN Y S, et al. High-performance hardware design for arithmetic coding in deep neural network-based image compression[J]. Journal of Electronics & Information Technology, 2025,47(9):3230-3240.
- [13] 夏桂书,何前,魏永超,等. 基于以太网与 FPGA 图像处理的边缘检测系统[J]. 仪表技术与传感器, 2023(10):56-59.
XIA G SH, HE Q, WEI Y CH, et al. Edge detection system based on ethernet and FPGA image processing[J]. Instrument Technique and Sensor, 2023(10):56-59.
- [14] 刘泽皓. 基于 FPGA 的一种改进 Canny 算法图像边缘检测实现和应用研究[D]. 盐城:盐城工学院, 2023.
LIU Z H. Implementation and application of an improved Canny algorithm for image edge detection based on FPGA[D]. Yancheng: Yancheng Institute of Technology, 2023.
- [15] 刘远哲,刘旭光,刘宇. 基于 FPGA 的 Sobel 图像边缘检测设计[J]. 电子产品世界, 2025,32(1):18-20,34.
LIU Y ZH, LIU X G, LIU Y. FPGA-Based design for Sobel image edge detection[J]. Electronic Engineering and Product World, 2025,32(1):18-20,34.
- [16] 周豪腾. 基于机器视觉的板带材表面缺陷检测与分类研究[D]. 秦皇岛:燕山大学, 2018.
ZHOU H T. Detection and classification of strip surface defects based on machine vision [D]. Qinhuangdao:Yanshan University, 2018.
- [17] 曹益,朱香平,张笑墨,等. 基于 FPGA 的低照度 EBCOMS 图像噪声处理算法[J]. 光子学报, 2024,53(11):266-280.
CAO Y, ZHU X P, ZHANG X M, et al. Noise image processing algorithm for a low-light EBCOMS acquisition system based on FPGA[J]. Acta Photonica

- Sinica, 2024, 53(11):266-280.
- [18] 祁欣, 陈剑锋, 罗伟林. 图像预处理算法的 FPGA 实现[J]. 国外电子测量技术, 2021, 40(2):102-107.
- QI X, CHEN J F, LUO W L. FPGA-based real-time image preprocessing system [J]. Foreign Electronic Measurement Technology, 2021, 40(2):102-107.
- [19] 叶惠娇, 冉全, 成果. FPGA 图像边缘检测系统的设计与实现[J]. 计算机应用与软件, 2018, 35(12):237-240, 302.
- YE H J, RAN Q, CHENG G. Design and implementation of FPGA image edge detecting system[J]. Computer Applications and Software, 2018, 35(12):237-240, 302.
- [20] 汤露. 基于 FPGA 的 Canny 算法硬件加速电路设计[J]. 工业控制计算机, 2023, 36(7):70-72, 76.
- TANG L. Hardware acceleration circuit design of real-time Canny algorithm based on FPGA[J]. Industrial Control Computer, 2023, 36(7):70-72, 76.
- [21] 段红燕, 邵豪, 张淑珍, 等. 一种基于 Canny 算子的图像边缘检测改进算法[J]. 上海交通大学学报, 2016, 50(12):1861-1865.
- DUAN H Y, SHAO H, ZHANG SH ZH, et al. An improved algorithm for image edge detection based on Canny operator [J]. Journal of Shanghai Jiaotong University, 2016, 50(12):1861-1865.

作者简介

陈南(通信作者), 本科, 主要研究方向为光学检测设备研发。

E-mail: nan.chen@emg-china.com