文章编号:1671-4598(2018)05-0001-04 DOI:10.16526/j.cnki.11—4762/tp.2018.05.001 中图分类号:TP274 文献标识码:A

基于 Linux 的激光焊保护镜镜面缺陷检测系统

王冠一，任永杰
（天津大学 精密仪器与光电子工程学院，天津 300100）

摘要：针对线上激光焊接的熔渣飞溅物粘附在保护镜片使激光光路受阻的问题，详细介绍了基于嵌入式 Linux 的视觉镜面缺陷检测系统的硬件及软件设计；提供了一种利用图像传感器对镜片进行视觉检测的解决方案；其中：MT9M031 为图像传感器，通过 V4L2 接口将采集图像映射到系统的用户空间并利用核心算法对图像进行坏点的检测；一旦检测到坏点，系统立刻经过 CAN 总线向机器人报警并终止线上作业；按照该方法设计的系统，经过实验验证效果良好；迄今尚无视觉检测保护镜片的有效方法，首度将视觉测量应用于该测量领域。

关键词：激光焊接；Linux；保护镜片；嵌入式；视觉检测；V4L2

Cover Lens Visual Inspection System Based on Linux in Laser Soldering

Wang Guanyi, Ren Yongjie
(Tianjin University, Tianjin 300100, China)

Abstract: Aimed to the problems of spattered fused dregs blocking the laser path on the cover len during on—line operation, the hardware and software design of a cover lens visual inspection system based on Linux in laser soldering was developed with the introduction. A solution with an imaging sensor to detect the lens was provided. MT9M031, as the imaging sensor, mapped images acquired to the user space by the interface of V4L2 and the system recognized the bad point in the image with the core algorithm. The system also warned the robot through CAN bus immediately after the recognition of bad points and stopped the on—line operation. The equipment developed is utilized in the experiment which improves the good performance of the visual inspecting quality. So far, there is no effective method for the visual inspection of cover lens and the proposed visual inspection is applied to the measuring area for the first time.

Keywords: laser soldering; Linux; cover lens; embedded; V4L2

0 引言

应用于现代焊装过程的激光焊接技术在线作业时会频频产生熔渣飞溅。飞溅物将会随机性地粘附在作业中的激光焊头保护镜片上。由于保护镜片上粘有金属熔溅物，激光光束的传递光路极有受阻的可能性，将造成焊体受热不均匀，产生坏焊的现象[1]。国际上的汽车厂商依靠人工定期更换保护镜片完成对保护镜片产生缺陷的预防，一些国内汽车厂商尝试加入气动喷嘴来预防飞溅粒子[2]，但均无法满足低成本、高利用率的要求。

镜面检测本身难度较大，迄今尚无视觉检测保护镜片的有效方法。针对以上需求，设计了一种基于嵌入式 Linux 的激光焊接保护镜片的视觉镜面缺陷检测系统。由于保护镜片上的飞溅粒子体积小、散布随机，所以需要高分辨率图像作为视觉样本；同时出于高效生产考虑，系统不能采用传统的字符设备和块设备的内存管理方式，而是选用高效快捷、高健壮性[3]的 V4L2 框架。以流的方式连续将高分辨率图像映射到 Linux 的用户空间并利用核心算法进行图像识别。如果检测到缺陷，系统通过 CAN 总线向机器人报警并终止线上作业。该系统摆脱传统工业现场的总机控制，可通过触摸屏进行人机交互功能，具有良好的便携性。

1 系统硬件设计

基于嵌入式 Linux 的激光焊接保护镜片的视觉镜面缺陷检测系统的硬件结构框图如图 1 所示。

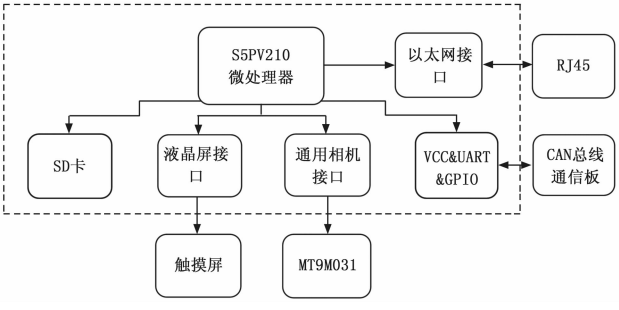

图 1 系统硬件结构框图

该系统硬件结构以 S5PV210 微处理器为核心、同时包括电源、DDR2、Flash、UART 接口、MT9M031 传感器、AT043TN24 液晶显示触摸屏、SD 卡插槽等。S5PV210 微处理器主要负责各模块间的控制和核心算法的运行。MT9M031 传感器通过以太网口将拍摄的图像映射到 Linux 的用户空间，并通过上层应用程序储存进 DDR2 和插入的 SD 卡中。检测系统与机器人通过 CAN 总线连接。

1.1 S5PV210 微处理器

S5PV210 是三星公司推出的一款适用于智能手机和平板电脑等多媒体设备的应用处理器，采用 ARM CortexTM—A8 内核，主频可达 1 GHz 并具有丰富的外围接口。

1.2 MT9M031 传感器

MT9M031 是由 Aptina 公司推出发行的一款 1/3 英寸高清图像传感器。激光焊接保护镜片的视觉信息通过 MT9M031 转化为数字图像。由于现代芯片的高度集成化，MT9M031 的 A/D 转换器集成在芯片内部，其外部电路设计十分简单。

收稿日期:2017-08-08； **修回日期**:2017-10-18。
基金项目:国家自然科学基金(51475329)。
作者简介:王冠一(1990-),男,辽宁鞍山人,硕士研究生,主要从事视觉测量方向的研究。

1.3 AT043TN24 液晶触摸屏

AT043TN24 为 CHIMEI INNOLUX 公司推出的一款 4.3 英寸的液晶触摸屏，常白显示，通透式，分辨率为 480(RGB) * 272。

1.4 CAN 转 UART 协议转换模块

由于激光焊接的工业机器人大多使用 CAN 总线来进行通讯，所以该系统配有 CAN 转 UART 协议转换模块来适应现场作业环境。该模块内部已经集成 DeviceNet 从站协议栈代码，可支持高达 128 字节 I/O 报文传输，适用于各种干扰强、实时性要求高的工业环境。

2 软件设计

2.1 软件开发环境

基于以上硬件平台，应用 Linux 操作系统作为嵌入式开发环境。Linux 是一款具有高时代性、高移植性的开源操作系统，可应用于个人 PC 机、服务器和嵌入式开发等[4]。较于其他应用于嵌入式开发的系统，如：WinCE、VxWorks，具有内核开源、强大的网络协议支持、

广泛的硬件支持等独有的特性。

在嵌入式 Linux 开发的过程中，用户需要对其内核进行修改、编译，同时增加对应硬件的驱动模块并在设备树中进行修改添加。主要操作如下：

1）裁剪和移植内核。只保留嵌入式设备需要的内核模块，节约空间；

2）移植和修改驱动模块。使得系统外设，如 MT9MO31 传感器，可被系统识别并访问；

3）修改和编译设备树。设备树文件是描述设备的属性和设置之间的关系，并且将设备树的配置最终通过驱动作用于实际的设备。该设计大大简化了驱动的移植与开发，当设备参数需要修改时，只需编译设备树文件，而不需重新编译内核。

2.2 硬件驱动程序

2.2.1 MT9M031 传感器驱动

MT9M031 驱动主要完成对相机的图像采集和配置的功能。图像从传感器发出后，通过 CMOS 控制器的调用，完成拍照。方案如图 2 所示。

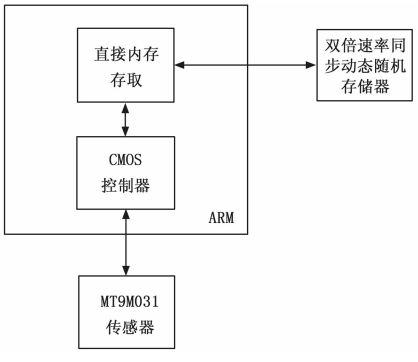

图 2 MT9M031 传感器驱动方案

首先在内核的设备树文件下添加 MT9M031 设备信息：

```
mt9m031@5d {
  compatible = "aptina,mt9m031";
  reg = <0x5d>
  port {
  mt9m031_1: endpoint {
  remote-endpoint = <&vpfe0_ep>;
  mclk-frequency = <50000000>;
  ...
```

i2c 的设备地址为 0x5d。compatible 属性的内容非常重要，该属性是和驱动匹配的唯一依据，" aptina" 代表厂家，" mt9m031" 代表 CMOS 传感器型号。" remote-endpoint" 代表和 CMOS 连接的外设，" mclk-frequency " 代表 CMOS 的输入时钟。

然后修改 vpfe 外设侧为：

```
&vpfe0 {
  ...
  port {
  vpfe0_ep: endpoint {
  remote-endpoint = <&mt9m031_1>;
  if_type = <2>;
  hdpol = <0>;
  vdpol = <0>;
  bus_width = <16>;
  ...
```

只需要修改 VPFE 远端连接的是 MT9M031 这款 CMOS 图像传感器即可。其中，if _ type 代表图像格式，2 代表 raw 格式。" bus _ width " 代表总线宽度。" hdpol " 代表行同步的极性，0 表示高电平有效，下降沿触发 DMA 中断，1 表示低电平有效，上升沿触发 DMA 中断。" vdpol " 代表场同步的极性，0 代表高电平有效，1 代表低电平有效。

然后是 MT9M031 驱动程序的移植。由于 Linux 内核中自带了 OmniVision 公司生产的 OV2659 传感器的驱动，通过修改其驱动文件完成 MT9M031 传感器的驱动移植。关键修改如：寄存器的读写函数、增加分辨率支持等。

首先将 OV2659 驱动文件中所有的字符串“ov2659”全部替换为“mt9m031”。由于替换众多，可以在 vim 编辑器的控制模式下使用全局命令：%s/ov2659/mt9m031/g 来进行全文档范围内的字符串替换。

下一步参照 MT9M031 的 datasheet 在该驱动文件中增加 MT9M031 的常用寄存器地址以及常量的宏定义，如： define MT9M031 _ CHIP _ ID _ REG 0x3000 等。

由于 ov2659 的 i2c 寄存器是 16 位地址，8 位数据，而 mt9m031 的 i2c 寄存器是 8 位地址，16 位数据，需要进行修改。修改如下：

```
static int mt9m031_write(struct i2c_client * client, u8 reg, u16 data){
return i2c_smbus_write_word_swapped(client, reg, data);}
static int mt9m031_read(struct i2c_client * client, u8 reg, u16 * val){
u16 * p_val = val;
* p_val = i2c_smbus_read_word_swapped(client, reg);
return 0;}
```

由于 Linux 启动之后，会自动匹配外接设备和驱动。一旦设备匹配成功后，操作系统会自动调用外设驱动中的 probe 函数。probe 函数将进行设备号的读取。如果读取到的设备号与 CMOS 的设备号（芯片 ID）相匹配，系统就会根据设备树中的配置信息对驱动设备进行初始化配置。配置结束以后，注册设备，驱动程序获得外接设备地址。在 mt9m031 _ probe 下的 mt9m031 _ detect _ sensor 函数中添加以下内容，进行复位 CMOS 芯片、关闭输出、使能 CMOS 芯片和读取 CMOS 芯片的版本号。该代码段中省略掉了错误处理部分：

```
    ret = mt9m031_write(client,MT9M031_RESET,1);
    mdelay(5);
    ret = mt9m031_write(client,MT9M031_RESET,0);
    ret = reg_write(client, MT9M031_OUTPUT_CONTROL, 0);
    ret = reg_write(client, MT9M031_CHIP_ENABLE, 1);
    ret = mt9m031_read(client,0x00, &ver);
    mt9m031->id = ver;
  if (mt9m031->id  MT9M031_ID)
  dev_info(&client->dev, "Found sensor,the version is %04x \
n",
  mt9m031->id);
  else{
    dev_err(&client->dev, "Sensor detection failed (%04X, %d)
\n",
  mt9m031->id, ret);
  ret = -ENODEV;}
```

由于在前面设备树文件下添加 MT9M031 设备信息中提到的 compatilble 属性是和驱动匹配的唯一依据，所以在 MT9M031 驱动文件中也需要添加和设备树中 compatilble 属性相同的信息。这样才可以完成设备和驱动的匹配工作。在 static const struct of _ device _ id ar0134 _ of _ match [] 中添加语句：{ . compatible = " aptina，mt9m031"，}，。同时在 static struct i2c _ driver ar0134 _ i2c _ driver 中的 . driver 项段中添加语句：. of _ match _ table = of _ match _ ptr（mt9 m031 _ of _ match），。

由于图像坏点的面积微小，系统要求采集 1024x1280 的灰度图像，为了支持这种格式的采集，需要使用分辨率为 1024x640 的 YUYV 图像格式进行替代。而 1024x640 的分辨率在原有驱动中并没有，需要在 mt9m031 _ framesizes 结构体中增加对这种分辨率的支持：

```
static const struct mt9m031_framesize mt9m031_framesizes[] = {
...
  { /* 1024x1280 */
. width = 640,
. height = 1024,
. regs = mt9m031_vga,
...  }
```

由于 V4L2 框架层与 MT9M031 驱动层之间存在着一个 CCDC Driver 层（VPFE 接口），所以此处需要涉及到 VPFE 接口驱动的修改。一般的视频采集涉及到的 VPFE 接口驱动是不需要修改的，针对该系统的视觉算法对图像的要求只需要修改内核源路径下的 vpfe. c 文件即可。该文件所处路径如下：

内核源路径/drivers/media/platform/ti-vpfe/vpfe. c

由于 Linux 内核中自带驱动的 OV2659 的像素时钟是上升沿锁存数据，而 MT9M031 的像素时钟是下降沿锁存数据，需要修改 pclk 的时钟极性。注释掉原有的 VPFE 配置语句：isif _ write（isif，cfg，ISIF _ VPFE _ CONFIG）；，将参数 cfg 改为 cfg | 0x0001，即将最低位置 1 表示下降沿锁存：isif _ write（isif，cfg | 0x0001，ISIF _ VPFE _ CONFIG）；

由于微处理器 SDK 中图像采集默认丢弃第一行像素，而 MT9M031 的第一行像素是有效的，不能丢弃，所以要修改 VPFE 接口驱动对采集图像第一行像素的处理。在 frm _ fmt ISIF _ FRMFMT _ INTERLACED 判断条件成立后的 if 语句段中注释掉语句 vert _ start + = 1；。vert _ start 变量表示 VPFE 接口采集图像的行起始标志。SDK 默认第一行像素丢弃，所有 vert _ start 标志自加 1，从第二行开始实时采集图像。注释掉该语句，VPFE 接口直接从第一行像素进行采集。

以上各个驱动程序修改完成后，将要进行 Linux 内核的重新编译。当在 Linux 源码树的顶层输入 make menuconfig 的时候，会出现内核编译的配置界面。该界面最终生成的就是依赖 Kconfig 文件（. config 文件），将编译的配置选项传递给 makefile，从而控制内核编译的过程。编辑内核源路径/drivers/media/i2c/Kconfig 文件，新增 MT9M031 的相关信息：

```
config VIDEO_MT9M031
tristate "APTINA MT9M031 sensor support"
depends on I2C && VIDEO_V4L2
select VIDEO_V4L2_SUBDEV_API
---help---
This is a V4L2 sensor-level driver for the APTINA
MT9M031 camera sensors.
```

与此同时接收内核编译配置选项的 makefile 也需要进行一些修改。打开控制驱动文件 mt9m031. c 编译的 makefile 文件，该文件路径为：内核源路径/drivers/media/i2c/Makefile。在原 ov2659 驱动控制编译语句下面添加 MT9M031 的驱动控制编译语句：

```
obj-(CONFIG_VIDEO_MT9M031) += mt9m031. o
```

最后将刚才修改的 MT9M031 的驱动文件的文件名改为 mt9m031. c，同时其对应的头文件名改为 mt9m031. h。在 Linux 源码树下输入 make ARCH=arm CROSS _ COMPILE=arm-linux-gnueabihf- menuconfig。选择 Device Drivers-> Multimedia support-> Encoders，decoders，sensors and other helper chips -> APTINA MT9M031 sensor support，选择该编译选项以后保存退出。然后开始编译内核，即可将 MT9M031 以上修改的所有驱动全部编译移植到新的内核镜像文件 zImage 文件。将该镜像文件和设备树编译后的 . dtb 文件通过 sd 卡烧进核心板，就完成了整个视频采集系统的内核移植和系统搭建工作。

2.2.2　AT043TN24 液晶触摸屏驱动

对于 480x272 的电阻屏的驱动，内核已经提供，但是刷新方式为横屏模式。由于图像的分辨率的关系，利用竖屏显示的支持方案效果最佳。只需要在 QT 编译的过程中，加入支持旋转的编译选项，将 QT 的库文件更新到文件系统中，然后在系统启动配置件中增加旋转屏幕的参数即可。

2.3　V4L2 图像采集程序

V4L2 框架下可以在程序中直接打开图像设备获取图像设备的能力，例如是否支持内存映射、设置图像格式（YUYV/UYVY）等。向驱动申请帧缓冲[5]，并且映射到用户空间。将映射到用户空间缓冲区入队列[6]，即可启动采集。启动采集以后，轮询是否采集到数据，当采集到以后，将帧缓冲出队列，对图像数据进行图像处理或者传输保存。出队列的缓冲使用完毕以后，继续入队列，等待下一次采集。当采集条件满足，例如采集到设定的帧数以后，停止采集，并且关闭设备。图像采集程序流程图如图 3 所示。

2.4　图像识别算法

该系统的图像识别核心算法是基于 OpenCV2 库函数实现的。当进入图像处理算法流程后，首先在图像内寻找保护镜片。因为同轴光会经过透射反射进入照相机，所以图像内的镜片应该为亮白色。如果找到则利用霍夫变换[7]来定位镜面区域，并判断是否发生定位区域越界。如果发生越界则提示镜片

不存在，未发生越界则取定位区域的外接矩形为感兴趣区域。在外接矩形感兴趣区域内首先进行保护镜片以外、外接矩形以内的区域置白，将隶属于以上区域中的像素点的像素值全部设置为 255。之后在保护镜片内部进行阈值设定和像素值二值化，目的是为了检索圆形镜片区域中的坏点轮廓。最后经过计算检索到的坏点像素面积以及比较其与预先人工设定的坏点面积的大小，得到最终的坏点判断。

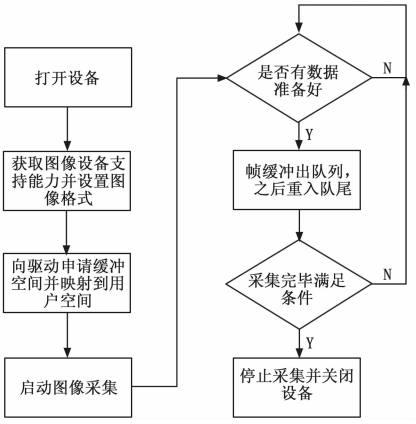

图 3 图像采集程序流程图

3 实验结果分析

3.1 实验数据

为了验证该基于嵌入式 Linux 的激光焊接保护镜片的视觉镜面缺陷检测系统的设计有效性，特进行了相关的实验操作和数据分析。实验共分为两种：1. 实验室条件下，对带有熔渣的缺陷保护镜片与无缺陷保护镜片混合检测；2. 在焊接现场对线上作业的保护镜片进行检测。实验数据如图 4 所示。

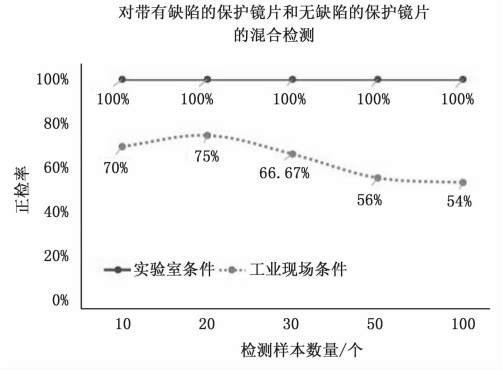

图 4 不同条件下对带有缺陷的保护镜片和无缺陷的保护镜片的混合检测

3.2 系统性能分析

该镜面缺陷检测系统的实物设备如图 5 所示。由实验数据可知，在实验室条件下，混合检测的正检率一直保持在 100%，系统搭建的有效性和核心算法的正确性得以验证。可是在工业现场中，熔渣的飞溅导致系统的照相机镜头上会粘附熔渣，直接影响到采图的客观真实性。因此随着检测样本的增加，正检率随意性变化。

3.3 有关照相机镜头除杂的补充实验

基于飞溅物粘附在照相机镜头上导致误检的问题，拟将一个通气导管放置在照相机镜头

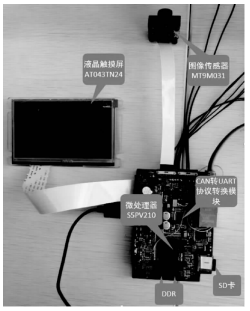

图 5 镜面缺陷检测系统的实物设备图

旁侧并时刻通气流斜向吹击照相机镜头。这样成功消除了飞溅物落附在照相机镜头上的可能性。增添通气除杂环节后，经核心算法处理后的镜片图像如图 6 所示。

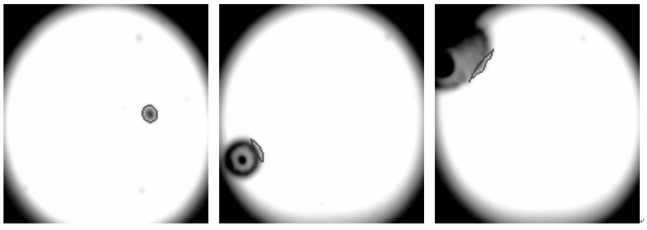

图 6 系统采集到的缺陷保护镜片的信息图像

4 结论

本设计是服务于现汽车生产线上激光焊接保护镜片的缺陷检测。首先采用了 S5PV210 微处理器，提高运算速度，降低采集能耗；然后选择 MT9M031 图像传感器并将保护镜片图像通过 DMA 映射到用户空间并使用 V4L2 框架以流的方式进行图像采集，这样可以快速高效地从相机处获取高分辨率的图像信息；设计 CAN 转 UART 协议转换模块和液晶触摸显示屏 AT043TN24，提高该系统的人机交互能力和现场适应度；最后基于 opencv2 函数库的核心算法编写：当检测出坏点时，红色线标记轮廓并报警。实验证明了本系统设计的有效性和算法的正确性，补充实验进一步完善了该系统对工业现场的适应性。

参考文献：

[1] 陈 勰. 启辰白车身激光焊接质量影响因素研究 [D]. 长沙：湖南大学，2011.
[2] 唐霞辉，朱海红，朱国富. 激光焊接专用气动喷嘴的研究 [J]. 激光技术，2000，24 (2)：95－98
[3] Michael H Schimek，Bill Dirks，Hans Verkuli. Video for Linux two API specification [EB/OL]. http：//v4l2spec. bytesex. org/ v4l2spec/v4l2. pdf. 2008.
[4] 邹自明，刘晓阳. 基于嵌入式 Linux 的结构化道路车道线识别系统的研究 [J]. 计算机测量与控制，2014，22 (4)：1260－1263
[5] 史 涛，裴海龙. 基于 Linux 的嵌入式实时视频跟踪系统 [J]. 计算机测量与控制，2014，22 (5)：1523－1526
[6] 徐 家，陈 奇. 基于 V4L2 的视频设备驱动开发 [J]. 计算机工程与设计，2010，31 (16)：3569－3572
[7] A. L. Fisher，P. T. Highnam. Computing the Hough transform on a scan line array processor [J]，IEEE Trans. Pattern Anal. Mach. Intel，1989，11 (3)：262－265.