计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(9):256−261 [doi: 10.15888/j.cnki.csa.006543]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 基于 ARM Cortex-A9 字符识别系统设计①

李前汭, 于力革

(江南大学 物联网工程学院, 无锡 214122)
通讯作者: 于力革, E-mail: yulige@163.com

**摘　要**: 本文搭建了以 ARM Cortex-A9 为核心的嵌入式字符分割识别处理平台, 研究了在 Linux 操作系统中交叉编译 OpenCV、Qt、BootLoader, 并移植相关驱动到嵌入式平台的方法; 基于经典算法使用 OpenCV 库函数实现了图像字符的分割和识别处理; 最后, 在嵌入式平台上移植了开发程序并进行了实验. 结果表明, 该系统能对图像字符进行很好的分割和识别.

**关键词**: 字符识别; 字符分割; ARM Cortex-A9; 图像处理

引用格式: 李前汭,于力革.基于 ARM Cortex-A9 字符识别系统设计.计算机系统应用,2018,27(9):256–261. http://www.c-s-a.org.cn/1003-3254/6543.html

## Design of Character Recognition Processing System Based on ARM Cortex-A9 Embedded Platform

LI Qian-Rui, YU Li-Ge

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

**Abstract**: In this study, an embedded character recognition processing platform based on ARM Cortex-A9 is set up. The method of cross compiling OpenCV, Qt, BootLoader in the Linux operating system and transplanting the related drive to the embedded platform are studied. Based on the classical algorithm, the image character segmentation and recognition processing is realized by using the OpenCV library function. At last, the development program is transplanted on the embedded platform and the experiment is carried out. The results show that the image character can be segmented and recognized well.

**Key words**: character recognition; character segmentation; ARM Cortex-A9; image processing

随着计算机技术和数字图像处理技术的发展, 机器视觉越来越广泛运用于实际生活中, 如通信领域、自动化生产等领域得到了越来越广泛的应用. ARM 嵌入式图像处理技术相对于基于 PC 机等传统的图像处理系统, 其具有小巧、灵活、实时性强、便携性好、低功耗、抗干扰强、程序固件化等优点, 得到了越来越广泛应用. 但也存在显而易见的缺点, 硬件方面如处理器的处理速度不够快, 以及主频较低、缓存比较小、软件方面设计难度大、开发难度高和专业性要求高等等. 针对上述优缺点, 本文在研究 ARM 嵌入式系统原理的基础上, 设计了一种基于 ARM Cortex-A9、OpenCV 函数库构建嵌入式图像处理系统来实现对文字图像的处理. 该系统包括硬件平台的构建和软件架构设计, 实现图像的采集、显示、存储及处理. 该系统采用 Qt 编程技术开发图形应用程序, 能实现图像实时显示和处理, 从而缩短程序开发周期和降低开发难度.

## 1　嵌入式图像处理平台硬件环境构建

本文硬件核心处理器是采用三星公司的基于 ARM Cortex-A9 四核的 Exynos4412 (Exynos4 Quad), 主频可达 1.6 GHz 处理器, 并拥有双通道 DDR3, 处理

① 收稿时间: 2018-01-25; 修改时间: 2018-02-27; 采用时间: 2018-03-08; csa 在线出版时间: 2018-08-16

器支持 1Mbyte L2cache, 具有可靠性高、稳定性好等特点. 同时平台采用 64 MB SDRAM、64 MB NandFlash 作为系统存储器来存储图像的处理结果, LCD 显示触摸屏显示采集图像和图像处理过程及结果等[1]. 摄像头选用的是与硬件开发平台配套的 Webeye2000 USB 摄像头来采集图像. 为了与 PC 机通信和向嵌入式平台烧录程序, 平台通过数据发送模块、JTAG 模块以及串口和网线接口连接到上位 PC 机, 组成了整个硬件开发平台, 这些硬件资源能够满足嵌入式文字识别系统的开发. 本文嵌入式图像处理系统的硬件结构图如图 1 所示.

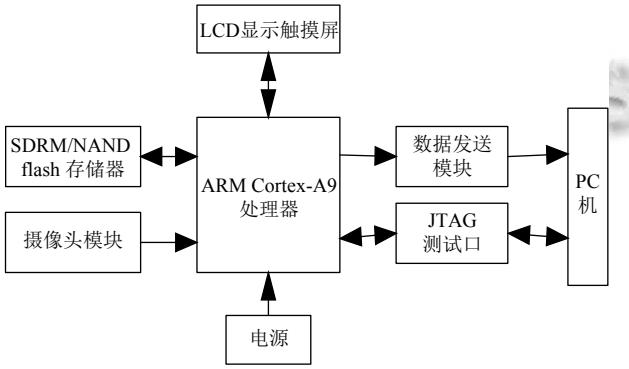


图 1 系统硬件结构图

## 2 嵌入式图像处理系统软件环境的构建

由于嵌入式平台上可利用的资源有限, 所以直接在平台上很难完成软件的开发和测试, 因此, 在嵌入式平台上开发相关应用程序时, 要构建交叉编译环境, 在 PC 机上完成相关程序开发和调试后, 通过交叉编译生成嵌入式平台可执行的文件. 嵌入式图像处理系统开发环境的构建主要包括 Bootloader 移植、操作系统的移植、摄像头驱动配置、Tslib 驱动移植和 OpenCV 的移植等.

### 2.1 操作系统移植

对于嵌入式开发系统平台构建的首要步骤是操作系统的选择和移植. 目前, 嵌入式 Linux 操作系统作为嵌入式主流操作系统, 其最大的特点是源码公开并且遵循 GPL 协议, 同时嵌入式 Linux 操作系统支持广泛的硬件, 可以同各种强大的应用软件及完善的设备驱动和开发工具融合. 嵌入式 Linux 操作系统移植的关键在于制作根文件系统以启动 Linux 内核, 制作根文件系统步骤如下:

(1) 建立 rootfs 目录, 并在该目录下建立 root、home、lib、dev、shin、proc、 tmp、var、etc、usr 等子目录, 并设置 rootfs 的可读写权限.

(2) 编译 busybox 文件, 将生成的 bin、sbin、linuxrc、etc 文件拷贝到 rootfs, 在 dev 里面建立新的结点 devA、devB. 执行 cramfsck -x rootfs rootfs. cramfs 命令, 生成 rootfs.ext4 根文件镜像, 将该文件拷入到开发板中, 完成根文件系统的制作.

### 2.2 Bootloader 的移植

Bootloader 是在目标板上电后, 运行于操作系统之前的一段小程序, 它的作用是引导加载应用程序. 能够用来初始化硬件配置、建立系统空间映像表, 从而建相匹配的系统软件和硬件环境[2], 为最终调用嵌入式图像处理平台的 Linux 系统做好准备. 修改 Makefile 文件, 进行相关宏定义修改, 主要步骤如下:

(1) 添加交叉编译库 include 文件路径到系统中, 即:

INCLUDE_DIR=/usr/local/arm/arm-2009q3/include/

(2) 添加交叉编译工具路径, 即:

CROSS_COMPILE=/usr/local/arm/arm-2009q3/bin/arm-none-linux- gnueabi-

(3) 交叉编译工具依赖文件, 即:

CROSS_COMPILE_ARM_GCC_LIBS=/usr/local/arm/arm-2009q3/lib/gcc/ linux-arm/

进行上述配置后, 执行 make menuconfig 和 make 生成二进制文件并烧录到开发板.

### 2.3 嵌入式平台摄像头驱动配置

进入 Linux 内核配置界面, 依次选择 Device Drivers→Multimediac Surport→Video capture adapters→V4L USB devices→USB Video Class 便可完成摄像头驱动模块的加载配置.

内核配置编译完成后会生成 zImage 内核镜像文件, Bootloader 会自动加载 zImage 到内核, 并跳转到 zImage 开始地址运行. 开发板上电, 在 Linux 超级终端 minicom 命令行中输入 root 命令, 会启动嵌入式平台 Linux 操作系统出现, 启动界面如图 2 所示.

### 2.4 Tslib 驱动移植

Tslib 是用来校准电阻式触摸屏开源软件库, 可以为触摸屏驱动获得的采样提供滤波、去抖、校准等功能, 经常用作触摸屏驱动的适配层, 在系统硬件的上层为应用提供接口. 移植步骤如下文.


图 2　嵌入式平台 Linux 启动界面

(1) 进入 Tslib 源码目录下, 配置交叉编译工具环境变量、生成路径, 进行编译之前的配置, 执行命令:

# ./autogen.sh

#./configure

--host=arm-none-linux-gnueabi ac_cv_func_malloc_0_nonnull=yes --cache-file=arm-linux.cache -prefix=/media/inand_ext4_1/tslib

(2) 完成 tslib 交叉编译后, 进入 tslib/etc 目录, 配置文件 tslib.conf, 把 module_raw_input 前面的“#”和空格去掉. 将 tslib 动态库移植到嵌入式开发板, 配置 tslib 在开发平台上的环境变量.

### 2.5　Qt 的驱动移植

Qt 是一种跨平台且面向对象的 C++图形用户界面应用程序开发框架. 它既可以开发 GUI 程序, 也可用于开发非 GUI 程序. Qt 使用一些特殊的代码生成扩展以及一些宏, 它易于扩展, 允许组件编程. 本文选用 Qt-4.8.5, 其在 Linux 嵌入式平台的移植步骤如下:

(1) 进入 embedded Linux 版本 qt-everywhere

-opensource-src-4.8.5 源码目录下, 执行命令, 进行编译之前的配置, 添加交叉编译工具 inux-arm 环境路径、添加-embedded arm 指定对 arm 平台 embedded 版本、添加-qt-mouse-tslib 和 tislib 的 lib 库路径为了指定使用 tslib 来驱动触摸屏.

(2) 编辑文件 mkspecs/qws/linux-arm-g++/

qmake.conf, 添加 tslib 动态库, 将 QMAKE_CC、QMAKE_CXX、QMAKE_LINK、QMAKE_LINK_SHLIB 的 arm-none-linux- gnueabi-gcc/g++后添加-lts, 将 QMAKE_AR、QMAKE_OBJCOPY、QMKE_STRIP 的 arm-none-linux- gnueabi-后面分别添加-ar cqs、-objcopy、-strip.

(3) 在终端 make 和 make install 后, 将经过交叉编译得到的 lib 库拷贝到嵌入式平台的根目录下, 配置 Qt 库在开发平台环境变量.

### 2.6　OpenCV 的移植

OpenCV 提供了大量的图像处理算法函数, 开发者可以在图像处理程序中直接调用算法并进行移植, 可以减少繁杂的开发任务. 本文移植的视觉库是 OpenCV-2.4.9, 移植步骤如下:

(1) 下载和安装 cmake-gui. 它是跨平台交叉编译工具, 能够生成 makefile 文件.

(2) 下载安装 OpenCV-2.4.9. 下载完成后解压到/root/tools 目录下, 切换到 root 用户下, 输入#cmake-gui 打开 cmake 图形界面, 输入 OpenCV 的源码路/root/tools/ OpenCV-2.4.9 和安装路径/user/local/arm-opencvlib, 点击 configure 和 next 按钮后, 添加 C/C++交叉编译工具, 并勾上 BUILD_JPEG、BUILD_PNG, 取消安装 tiff, 修改 CMAKE_INSTALL_PREFIX 值, 点击 Generate 按钮生成 Makefile 文件. 进入到 arm-opencvlib 目录下 CmakeCache.txt 文件, 在 CMAKE_EXE_LINKER_FLAGS:STRING=后加上-lpthread –lrt; 在终端输入 make 和 make install 完成 OpenCV-2.4.9 编译安装.

(3) 进入 arm-OpenCV 目录, 将编译连接得到的 lib、include、share 文件拷贝到开发板的/user/local 目录下, 并配置嵌入式板的 OpenCV 库的环境路径.

## 3　嵌入式文本字符识别程序开发

本文设计了对汉英混排字符文本文档图像的字符识别程序, 文字识别处理流程如图 3 所示.


图 3　文字识别处理流程

### 3.1　图像预处理

由于在图像的获取过程中, 拍摄的角度及相机的振动和拍摄的高低远近变化无常, 再加上光线的变化, 本身曝光时间, 这些因素都会影响所采集图像的质量. 为了使在后续对图像中信息更好识别获取, 必须使区域特征更加突出, 目标信息更加明显, 故在对图像进行信息识别之前, 需要经过一系列的预处理来对图像进行补偿[3], 以减少识别的错误率. 图像预处理工作依次为对采集到的图像进行图像去噪、灰度化处理、边缘处理、图像二值化等.

为了在去噪过程中, 不损坏图像中字符的轮廓和边缘等信息, 本文采用了中值滤波法进行滤波, 在消除

噪声的同时不破坏图像的细节. 在灰度化处理方面, 本文选择颜色通道的最大值作为灰度值, 即:

$$Gray(i,j)=\max\{R(i,j),G(i,j),B(i,j)\}\quad 0\le i\le W,\ 0\le j\le H \tag{1}$$

其中, $W$、$H$ 为图像的宽和高.

另外, 采用 Sobel 算子进行文字的边缘检测, OTSU 算法进行图像的二值化, 预处理效果如图 4 所示.


图 4 预处理效果

### 3.2 文本字符图像分割

在预处理完成后, 利用文本图像版面一些特点, 首先采用水平投影进行行分割, 其次基于字符连通域的方法对二值图像进行字符连通域标记[4,5], 以及将中文区域和英文区域进行分离, 最后已分割出来文字进行合并和粘连分割, 处理步骤如下:

(1) 行分割 对于印刷字体来说, 一般行界区分都很明显, 即文本图像行与行之间的空白空隙是固定的, 而且要小于字符行的高度, 为了更好的处理, 可以通过检测空白区域, 可以确定图像中的各行边界, 本文根据这一特征采用水平的投影算法[6]来进行行割. 目前, 对文本字符分割采用投影法效果明显. 它是把图像具体化为一个 A×B 的矩阵 $f(i,j)$, 对每一行的投影算法如下:

$$V(j)=\sum_{i=0}^{N} f(i,j),\ j=0,1,2,\cdots,N \tag{2}$$

其中, 投影值为 0 的点是字符间的空白. 将第一个 $V(j)\neq 0$ 的点标记为 $V_a$, 并从 $V_a$ 开始, 分割程序从上到下或从左到右扫描每一行或列, 当第二次遇到 $V(j)=0$ 的点时, 将该点标记为 $V_b$, 此时扫描停止, 这两点之间视为一行字符, 使用这种方法循环至行的结尾.

(2) 字符连通域标记 在完成行分割后, 就可以进行字符分割. 除了行与行之间, 在同一行中的字符之间也是存在空白的, 可以利用这些空白, 可以确定字符之间的边界. 可以利用该特性, 对图像不同行区域的字符进行连通域标记, 并定义连通域类如下:

```
struct UnionArea
{ UnionArea* up //指向上一个连通域的指针
Point Rt //连通域的左上角顶点
Point Lb //连通域的右下角顶点
UnionArea* next //指向下一个连通域的指针
}
```

对文本进行初步切分, 大多数文字可以被切分出来, 但仍然有少许文字不能被正确切分出来, 如左右结构汉字、上下结构汉字、粘连字符和交叉包含结构的字符会形成独自连通域, 如图 5 所示, 因而对初步切分结果进行合并处理是很有必要的[7]. 经过试验表明, 采取以下几个步骤能够较为有效地切分出各文字.




图 5 连通域初步标记及初步切分

1) 区域分割: 根据文本图像中同一水平线上, 同类文字之间中心间距是均匀且固定的, 因此可以利用这种特性来判断一些字符是否为英文字符或中文字符[8,9]. 但是当不同类文字混合出现在一行的时候, 对等间距性产生破坏[10]. 文本中英文字符的长和宽远远小于中文字符, 可以利用字符的长和宽特性来分离出中英文字符区域[11]. 因为英文字符和数字字符有类似的宽度和周期, 所以本文在区域分离的时候, 将数字考虑在英文区域内. 但是对于标点符号来说, 其长宽比相对来说要小很多, 可以依此快速分离出标点符号.

2) 字符合并: 首先对连通域进行行排列, 以连通域的中心点为特征, 对每行的连通域按照中心点列坐标进行排序. 其次, 合并分两步进行, 根据上下结构和交叉包含连通域的中心点的横纵坐标都在最大连通域的范围内, 将完全包含关系和上下结构关系的连通域合并[12]. 最后, 以连通域中心点为中心, 以 $m$ 倍×平均字宽作为搜索半径进行左右搜索; 如果合并后的字宽不超过 $n$ 倍×平均字宽, 则进行合并 (其中 $m$、$n$ 为系数), 并结合左右结构字符的内部间隙小于字与字之间的距离进行合并.

3) 粘连分割: 首先, 求解平均字宽. 由于连通域大多是已经切分正确的文字, 因而可以通过统计连通域的宽度来得到一个参考的平均字宽[13]. 为了使得统计的平均字宽更接近真值, 需要舍弃或合并某些连通域进行多次平均求解, 会得到接近实际的平均字宽. 其次, 为了使粘连的字符或部件分离, 扫描结点, 判断连通域的宽度是否超过平均字宽一定程度. 如果超过, 则返回到层次连通域表中上一个层次得到多个连通域. 合并

和粘连切分如图 6 所示.

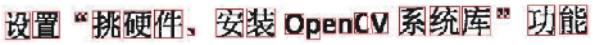


图 6 合并处理与粘连切分

## 4 字符特征提取与训练

由于汉字数量较大, 本文选取了 200 个宋体汉字作为训练样本, 100 个宋体汉字作为测试样本. 分别训练常用汉字、英文和标点模板, 形成样本库. 具体步骤如下: 首先将输入只有单个字符的图像, 进行中值滤波、二值化、归一化和特征提取等处理; 然后把提取出来的特征向量值存入自动生成的 HOG_SVM_CHIN.xml、HOG_SNM_DATA.xml、HOG_SVM_CHAR.xml 文件. 最后文件的命名时把手动输入指定的字符转化为对应的微软标准编码作为文件的名字, 后缀名为.xml 数据文件, 并把文件保存在指定的路径下.

## 5 实验测试

(1) tslib 测试, 把 tslib.tar.gz 拷贝到开发平台, 并解压到/media/inand_ext4_1 目录下, 运行 ts_calibrate 触摸校准程序, 触摸校准完成后会在/etc 目录下生成 pointercal 文件. 接着可以运行其他命令 (ts_harvest、ts_print、ts_print_raw、ts_test) 进行触摸测试.

(2) Qt 测试, 把 QtEmbedded-4.8.5-arm.tar.gz 拷贝到开发平台, 解压到/media/inand_ext4_1 目录下, 进入/media/inand_ext4_1/QtEmbedded-4.8.5-arm/demos/embedded/fluidlauncher, 运行 fluidlauncher 启动器程序, 触摸屏上将启动 Qt 图形界面, 并且可以用触摸操作.

(3) 编写开机运行 Tslib 和 Qt 启动脚本, 让系统开机就自动运行 Qt 图形界面程序, 需要在系统的启动文件里加上执行指令, 修改文件文件/usr/etc/rc.local.

(4) 在平台下安装 Qt 插件库, 在/usr/local 下安装支持 jpeg、png 等图片格式的插件库.

(5) 将已经训练好的字符库.xml 文件拷贝到嵌入式平台的/bin/目录下.

(6) 程序移植, 完成字符处理程序设计后, 使用交叉编译器对已经调试好的源程序编译, 生成可以在开发平台运行的可执行文件 QTlmage, 并将该文件移植到开发平台下的/bin/目录.

启动系统, 应用程序自动运行, 在嵌入式平台上看到程序运行效果如图 7 所示.

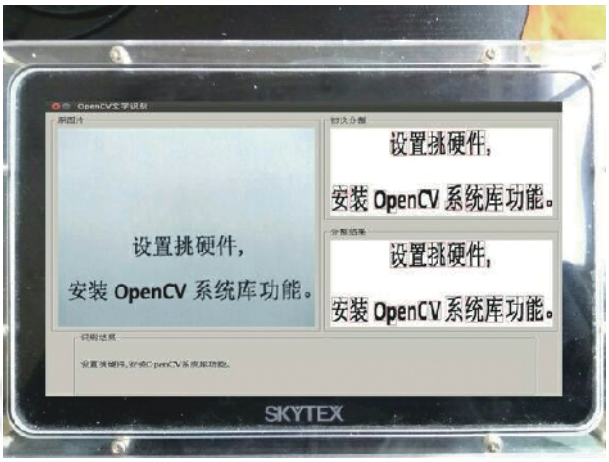


图 7 嵌入式平台 OCR 字符识别效果

## 6 结论与展望

本文研究了基于嵌入式平台混合字符识别系统, 给出了嵌入式系统开发环境的构建步骤, 主要包括: Bootloader 移植、摄像头驱动配置、Tslib 驱动移植和 OpenCV 的移植等, 开发了基于 ARM Cortex-A9 嵌入式图像字符识别程序, 并将程序移植到嵌入式平台运行, 从运行结果可知, 编译程序已经在嵌入式平台正常运行, 移植的 OpenCV 和 Qt 嵌入式图像处理系统具有较高的稳定性和可行性, 同时该系统也能对图像字符进行很好的分割和识别. 但在拍摄图片时, 图片的质量受多种因素的干扰, 字符分割时有时会产生错误, 导致识别出现误差. 因此, 下一步还需要改进算法, 来提高字符的识别.

### 参考文献

1 深圳市海天雄电子有限公司. CES-KY4412 科研开发平台产品手册. 深圳: 深圳市海天雄电子有限公司, 2015.

2 袁秋林. 基于 ARM9 平台的嵌入式 Linux 系统的移植开发及应用[硕士学位论文]. 兰州: 西北师范大学, 2009.

3 刘刚, 黄襄念, 符清芳, 等. 购物小票图像识别预处理算法的研究. 计算机时代, 2016, (4): 21–24.

4 于明, 郭佥, 王栋壮, 等. 改进的基于连通域的版面分割方法. 计算机工程与应用, 2013, 49(17): 195–198. [doi: 10.3778/j.issn.1002-8331.1111-0549]

5 孙婷. 基于连通域的中英文混排扭曲图像校正研究[硕士学位论文]. 北京: 北方工业大学, 2016.

6 陈艳, 孙羽菲, 张玉志. 基于连通域的汉字切分技术研究. 计算机应用研究, 2005, (6): 246–248. [doi: 10.3969/j.issn.1001-3695.2005.06.088]
7 李俊. 印刷体文字识别系统的研究与实现[硕士学位论文]. 成都: 电子科技大学, 2011.
8 Xu L, Yin F, Wang QF, *et al*. Touching character separation in Chinese handwriting using visibility-based foreground analysis. 2011 International Conference on Document Analysis and Recognition. Beijing, China. 2011. 859–863. [doi: 10.1109/ICDAR.2011.176]
9 王恺, 王庆人. 中英文混合文章识别问题. 软件学报, 2005, 16(5): 786–798.
10 索玉秀. 基于 OCR 技术的名片识别方法研究硕士学位论文. 哈尔滨: 哈尔滨理工大学, 2015.
11 安艳辉, 董五洲. 基于识别反馈的粘连字符切分方法研究. 河北省科学院学报, 2008, 25(2): 32–35. [doi: 10.3969/j.issn.1001-9383.2008.02.008]
12 任荣梓, 高航. 基于反馈合并的中英文混排版面 OCR 技术研究. 计算机技术与发展, 2017, 27(3): 39–43.
13 罗佳. 一种对粘连英文字符串的快速切分算法研究. 计算机技术与发展, 2014, 24(8): 59–62.