

·信号与信息处理·

## 基于 MiniGUI/Embedded 的图像处理技术

周培源

(西安电子科技大学软件学院, 陕西 西安 710000)

**摘要:**通过对嵌入式系统下数字图像处理存在的不足及 Windows 下的优势分析,提出了一种基于 MiniGUI/Embedded 的图像处理方法。介绍了 MiniGUI/Embedded 软硬件开发平台,说明了 MiniGUI/Embedded 开发环境配置,阐述了图像处理系统设计要点,以及静态或动态链接库的创建。借鉴 Windows 下的图像处理技术,将该技术移植到以 MiniGUI 为平台的嵌入式系统。结果表明,基于 MiniGUI/Embedded 的图像处理技术,方法可行,效果良好。

**关键词:**图像处理; MiniGUI/Embedded; 嵌入式系统; 图像处理算法

中图分类号: T391.41

文献标识码: A

文章编号: 1673-1255(2011)04-0064-05

## Image Processing Technology Based on MiniGUI/Embedded

ZHOU Pei-yuan

(College of Software, Xidian University, Xi'an 710000, China)

**Abstract:** By analyzing the weakness of the digital image processing in embedded systems and the advantage in Windows, a method of the image processing technology based on MiniGUI/Embedded is proposed. The software/hardware development platform and the development environment configuration about MiniGUI/Embedded are introduced. The design points about image processing system and the creation of the static or dynamic link library are expatiated. By drawing on the experience of the image processing technology in Windows and this technology is transplanted into MiniGUI embedded systems. The results show that this method of the image processing technology based on MiniGUI/Embedded is feasible and can get satisfied results.

**Key words:** image processing; MiniGUI/Embedded; embedded systems; image processing algorithm

目前已有许多专门针对 Linux 的嵌入式 GUI 系统, MiniGUI 是目前比较成熟的系统之一, 它是一个跨平台的、基于 SVGALib 及 Linux Thread 库的、多窗口 GUI 支持系统; 虽然不具备 Windows 平台上的 MFC、OWL、ATL 强大的功能, 但 MiniGUI/Embedded 具有优良的跨平台特性、轻型且占用资源少、高性能、高可靠性以及可配置性<sup>[1]</sup>, 越来越多地应用在中高端手机、WiFi 手机、PDA 类产品、工业自动化控制等高端设备中。而传统的图像显示与处理系统由于缺乏第三方软件开发的支持, 在高端设备中应用很少, 并且以 Windows 平台为代表的 MFC 所开发

的图像处理技术在上述产品中应用更是少之又少。因此, 如何将 Windows 平台上 MFC 开发的、成熟的图像处理技术移植到嵌入式 MiniGUI/Embedded 中, 成了目前亟待解决的问题。以下将结合两者优点, 将 Window 下图像处理技术移植到以 MiniGUI 为平台的嵌入式系统中, 内容主要包括编译环境的建立、文件系统设计、功能模块化管理、源文件中算法分析、调试中关键问题的解决方法, 并在 Virtual framebuffer 上调试运行以验证设计的有效性。

收稿日期: 2011-06-11

作者简介: 周培源(1988-), 男, 辽宁锦州人, 学士, 主要专业方向为软件工程。

# 1 MiniGUI/Embedded 开发平台

## 1.1 MiniGUI/Embedded 硬件平台

嵌入式硬件系统选用 S3C2410 开发板,以 Samsung 公司的处理器为核心,其内部集成 32 位 ARM920T 处理器硬件内核,标准频率为 203 MHz,平台的硬件包括 LCD 显示屏和主板两部分,系统扩展 64 MB NandFlash、2 MB NorFlash、64 MB SDRAM 和主/从 USB 等<sup>[2-3]</sup>。S3C2410 开发板平台通过串口、JTAG 并口、网口与宿主机通信,图像处理界面通过 LCD 显示,通过触摸屏/鼠标进行控制。系统的硬件平台体系结构及 S3C2410 开发板如图 1 和图 2 所示。

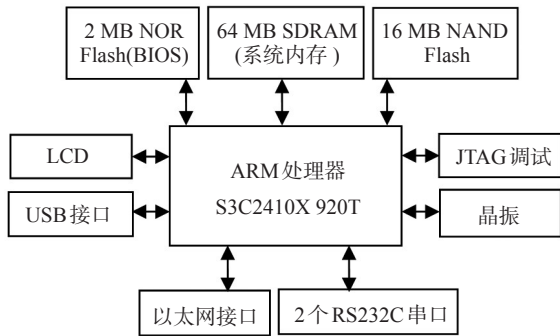


图1 硬件平台体系结构

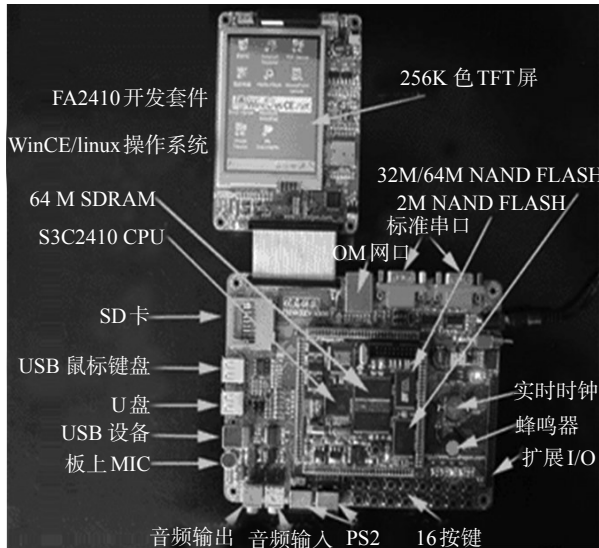


图2 S3C2410 开发板

## 1.2 MiniGUI/Embedded 软件平台

PC 机安装 Linux Redhat9.0 操作系统作为宿主机开发环境,通过 minicom 进行程序调试。S3C2410 目标开发板上移植 ARM-Linux,并且,该

开发板装有 Minigui1.6.10 环境,以便将 MiniGUI/Embedded 开发的应用程序发布到嵌入式 Linux 的文件系统上。

# 2 MiniGUI/Embedded 平台下的软件开发

## 2.1 MiniGUI/Embedded 开发环境配置

首先安装编译器 armv4l-unknown-linux-gcc,分别安装 libminigui-1.6.10.tar.gz, minigui-res-1.6.10.tar.gz,qvfb-1.1.tar.gz,激活 FrameBuffer,选择 qvfb 运行模式,配置过程中选择 640×480 "VGA" Depth 32 bit。特别注意的是在宿主机上编译 MiniGUI 应用程序,需加上 -lminigui-pithead-lm-lz-lmgext-lttf-lpng-ljpeg 链接选项,用于支持绘图函数库、线程库、MiniGUI 核心库、png 图形支持库等。

## 2.2 图像处理系统设计

系统的主要设计目标是将 Windows 平台下成熟的图像处理算法移植到 MiniGUI 平台,可以实现对 bmp 位图的显示和处理,如图片打开、保存、几何变换、图像检测与分割、滤波以及形态学处理等。首先,创建文件系统,包括 include 文件夹,obj 文件夹,src 文件夹,res 文件夹。其次,需要编写 3 种类型文件:头文件(.h)、实现文件(.c)和主函数文件(main.c)。整个文件系统的设计及编译过程如图 3 所示。

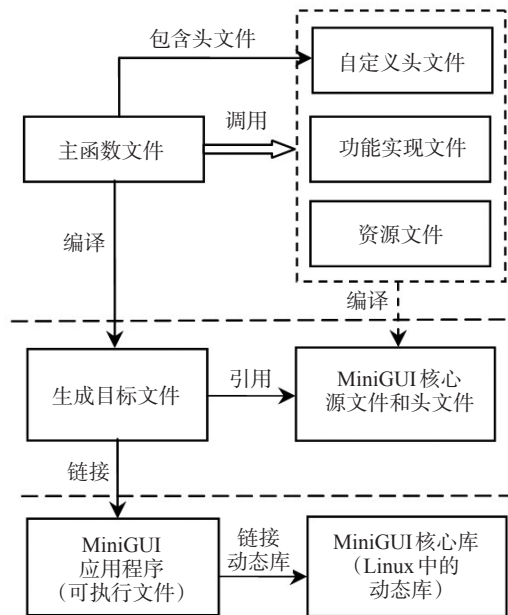


图3 MiniGUI/Embedded 文件系统设计和编译

### 2.2.1 头文件编写

在嵌入式系统中,由于没有 Windows MFC 系统所具有的强大的图像操作支持,因此,对图像操作所需要的结构、变量以及大部分算法都要由用户定义。BMPMAP.h 头文件包含了 bmp 位图所要定义的结构,包括位图文件头 BITMAPFILEHEADER、位图信息头 BITMAPINFOHEADER、调色板 Palette、以及实际位图数据 ImageData<sup>[3]</sup>。MAINGUI.h 包含了菜单、消息、自定义控件的 ID 标识,同时涵盖图像处理过程的宏定义、函数声明等。

### 2.2.2 图像处理算法文件编写

在 Windows 下,MFC 对图像处理提供了大量的函数,并且有很多成熟的图像处理算法。本系统在移植这些算法的过程中,采用了模块化设计,依据软件工程中高内聚低耦合原则,将每个具体的算法分成单个模块,使模块内部包含具体的功能实现,而模块之间尽量降低关联性。模块划分大致如下:图像预处理模块——将图像的位图格式转换成图像处理所需要的数据信息;图像处理模块——使用 Windows 下成熟的图像处理算法实现对图像的各种操作;图像保存模块——对处理后的图像进行保存。整个系统的模块构造如图 4 所示。

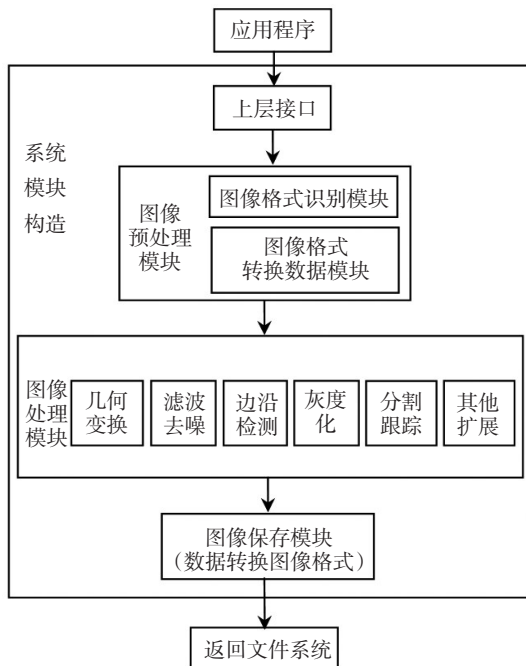


图4 系统模块结构图

下面以图像分割算法为例,具体介绍移植方法。首先,由图像预处理模块,将位图格式转化为 Windows 相应的格式。然后,按照 Windows 下的图

像处理算法编写 Linux 下的图像处理算法(否则,图像处理后的效果差异较大),之后可对算法进行改进。下面介绍一种迭代求解图像最佳分割阈值算法<sup>[4]</sup>:

(1) 首先选取图像灰度范围的中值作为阈值  $T$  的初始估计值;

(2) 根据阈值  $T_i$  将图像分割成目标和背景两部分,分别求出两部分的平均灰度值  $\mu_1$  和  $\mu_2$ , 计算公式分别为

$$\mu_1 = \frac{\sum_{k=0}^{T_i} n_k \cdot k}{\sum_{k=0}^{T_i} n_k} \quad (1)$$

$$\mu_2 = \frac{\sum_{k=T_{i+1}}^{L-1} n_k \cdot k}{\sum_{k=T_{i+1}}^{L-1} n_k} \quad (2)$$

其中  $n_k$  是灰度为  $k$  的像素点个数。

(3) 计算新的阈值

$$T_{i+1} = \frac{\mu_1 + \mu_2}{2} \quad (3)$$

重复步骤 2、3,直到得到的阈值  $T$  值之差小于一个事先定义参数为止。最后,经图像保存模块,将处理后的图像保存即可。

在整个移植过程中,算法的思想和 Windows 要保持一致,并且大部分代码与 Windows 相同,其中,需要特别说明的是,在处理图像数据信息时,Windows 下需要用 Globallock() 全局内存管理 API 函数来锁定内存,以便从指定的内存块进行读写操作。而在 Linux 下没有 Globallock() 函数,因此使用 malloc() 函数代替<sup>[5]</sup>。而全局内存管理函数 Globallock() 比其他内存管理函数慢,而且,考虑到兼容性,该函数仍然用于 Windows 下 DDE 和剪切板函数,并且不能提供更多的特征<sup>[6]</sup>。这样,以 MiniGUI 为平台的图像处理系统,在速度上比 Windows 下要快。

### 2.2.3 主函数编写

MiniGUI 是一个基于事件驱动的图形用户界面支持系统,所以每一个 MiniGUI 应用程序都包含有一个主事件循环,它不停地、反复地检查是否有用户事件发生。当用户进行窗口操作时,如移动窗口、触发菜单、单击鼠标、键盘输入等,这些动作就会触发一个相应的“事件”,而应用程序把这些事件

交予该窗口相关联的窗口过程函数处理<sup>[7]</sup>。

在 main.c 主函数中,就是依据上述的事件和消息驱动机制,也就是事件通过消息机制来传递,当

事件产生时,MiniGUI将该事件转换为一个消息,并将消息发送到消息的循环队列之中。主函数中事件消息处理过程大致如图5所示。

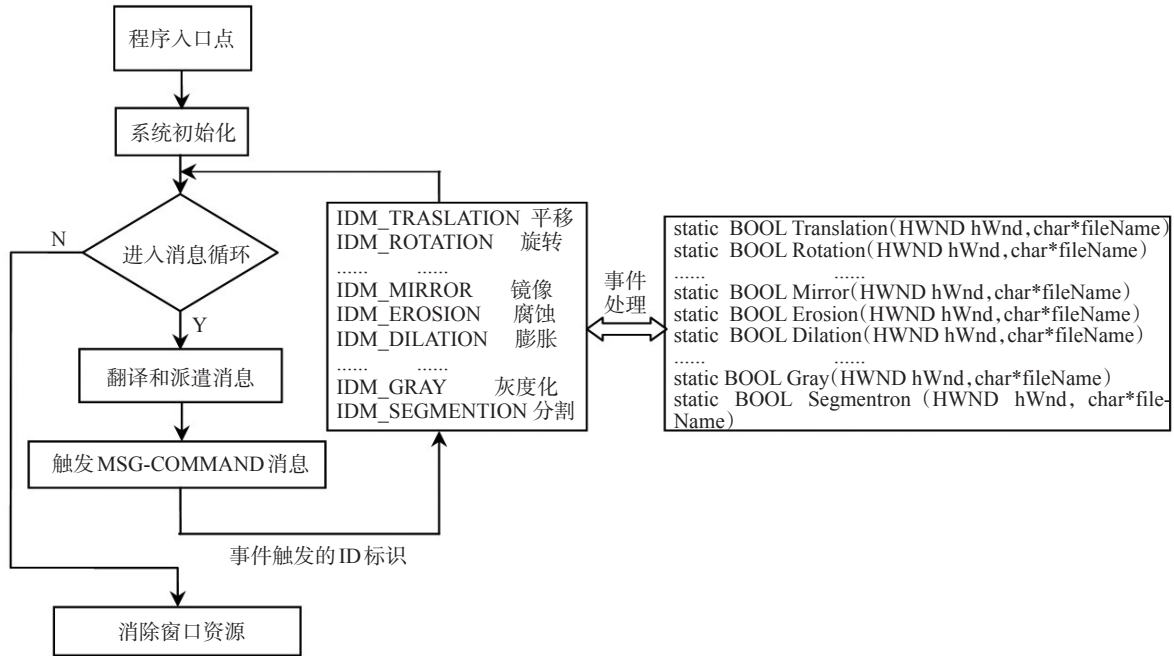


图5 主函数中事件消息处理过程

在 main.c 主函数文件中,还包括主窗口的相关设置、菜单的定制、工具栏的定制。设置图像显示的坐标位置大小等。

### 2.2.4 Virtual framebuffer上图像处理系统的交互界面

在 Linux 系统的 Virtual frambuffer 上运行可执行文件,以避免在开发板上进行繁琐的修改和调试。首先分别用 Autonconfig 和 make 工具生成相应的 Makefile 文件、目标文件(.o)及可执行文件,运行 qvfb,最后,执行可执行文件即可。

至此,可以在 pc 机的 framebuffer 上显示主界面,并且在这个基础上,反复修改调试图像处理文件源代码以达到在 Windows 环境下同样的效果,进而,使整个系统达到理想效果,该系统交互界面如图6所示。

### 2.3 创建静态或动态链接库

在 Windows MFC 下,有些动态库是 API 支持的,提高了开发的效率和质量。借鉴该项技术,将一些公用的函数制作成 linux 函数库(静态库和动态库)。静态库——在程序编译时会被链接到目标代

码中,程序运行时不再需要的库文件。动态库——在程序编译时不被链接到目标代码中,在程序运行时才被载入的库文件。这样可根据需要将 BMP-MAP.h 和 bmpOperation.c 分别做成静态库文件和动态库文件,以提高系统的运行效率<sup>[8]</sup>。



图6 用户交互界面

## 3 实验结果

对图7原始图像,分别在 Windows 下运行图像算法程序和 Virtual framebuffer 上执行 MiniGUI 可执行文件。图像处理算法包括:灰度化、分割等,最后得到的图像结果如图8、图9所示。从中可以看



出2种平台下图像处理效果是基本相同的。



图7 原始图像



图8 Windows下阈值分割图像



图9 MiniGUI下阈值分割图像

## 4 结束语

实验表明,基于MiniGUI/Embedded系统的图像处理技术,方法可行,效果良好。该系统用户界面友好、成本低、处理速度快、可移植性强。系统可以根据需要自行定制改进,以适用于嵌入式卫星图像采集跟踪、医学图像分析、手机照片处理、PDA图片浏览与处理等多种应用领域。

## 参考文献

- [1] 北京飞漫软件技术有限公司. MiniGUITEchWhitePaper-1.3C[M]. 北京:北京飞漫软件技术有限公司,2003:3-6.
- [2] Samsung Electronics S3C2410A - 200MHz & 266MHz 32-Bit RISC Microprocessor User's Manual[S]. Revision 1.0 (March 2004) Yongin-City: Samsung Electronics, 2004:31-35.
- [3] 全青松,余厚全. 嵌入式视频监控终端的设计与实现[J]. 光电技术应用, 2010,25(6):53-55.
- [4] 谢凤英,赵丹培. Visual C++数字图像处理[M].北京:电子工业出版社,2008.
- [5] 广树建. 新编C/C++程序设计教程[M].广州:华南理工大学出版社,2011.
- [6] Jeffrey Richter. WINDOWS核心编程[M]. 5版. 黄隴,李虎. 北京:机械工业出版社,2008.
- [7] 苏东. 主流ARM嵌入式系统设计技术与实例精解[M]. 北京:电子工业出版社,2007.
- [8] 杨宗德. 嵌入式ARM系统原理与实例开发[M].北京:北京大学出版社,2007.

(上接第55页)

- electrochemical and chemical dissolution of wafers[J]. Appl. Phys. Lett, 1990, 57(10):1046-1048.
- [2] Velumani S, Ascencio J A. Formation of ZnS nanorods by simple evaporation technique[J]. Applied Physics A: Materials Science & Processing,2004,79(1):153-156.
  - [3] Wang Cai-Feng, Li Qing-Shan, Hu Bo, et al. The effect of annealing on structural, optical and electrical properties of ZnS/porous silicon composites[J]. Chinese Physics B, 2009, 18 (6):2610-2614.
  - [4] Yano S, Schroeder R, Sakia H, et al. High-electric-field photocurrent in thin-film ZnS formed by pulsed-laser deposition [J]. Appl. Phys. Lett, 2003,82(13):2026-2028.
  - [5] Nasrallah T Ben, Amlouk M, Bernede J C, et al. Structure and morphology of sprayed ZnS thin films [J]. Phys. Stat. Sol., 2004, 201(14):3070-3076.
  - [6] Wang Caifeng, Li Qingshan, Lv Lei, et al. Structural, Optical and Electrical Properties of ZnS/ Porous Silicon Heterostructures [J]. Chinese Physics Letters, 2007, 24 (3):825-827.
  - [7] XU Dongsheng, GUO Guolin, GUI Linlin. Controlling growth and field emission property of aligned carbon nanotubes on porous silicon [J]. Appl. Phys. Lett, 1999, 75(4):481-484.
  - [8] Morozova N K, Karetnikov I A, Plotnichenko V G, et al. Transformation of luminescence centers in CVD ZnS films subjected to a high hydrostatic pressure[J]. Semiconductors, 2004, 38(1):36-41.
  - [9] Prokes S M, Freitas Jr. J A, Searson P C. Microluminescence depth profiles and annealing effects in porous silicon[J]. Appl. Phys. Lett., 1992,60(26):3295-3297.