



2012 International Conference on Medical Physics and Biomedical Engineering

## Transplant Image Processing Technology under Windows into the Platform Based on MiniGUI

Lan Gan<sup>1</sup>, Xu Zhang<sup>1,\*</sup>, Wenya Lv<sup>1</sup>, Jia Yu<sup>1</sup>

1. Information Engineering School, East China Jiaotong University, Nanchang 330013, China

E-mail: [gl7046798@yahoo.com.cn](mailto:gl7046798@yahoo.com.cn)

\*E-mail: [zhangxu\\_happy@126.com](mailto:zhangxu_happy@126.com)

E-mail: [lvwenya123@163.com](mailto:lvwenya123@163.com)

---

### Abstract

MFC has a large number of digital image processing-related API functions, object-oriented and class mechanisms which provides image processing technology strong support in Windows. But in embedded systems, image processing technology dues to the restrictions of hardware and software do not have the environment of MFC in Windows. Therefore, this paper draws on the experience of image processing technology of Windows and transplants it into MiniGUI embedded systems. The results show that MiniGUI/Embedded graphical user interface applications about image processing which used in embedded image processing system can be good results.

© 2012 Published by Elsevier B.V. Selection and/or peer review under responsibility of ICMPBE International Committee.

Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/4.0/).

Index Terms—Embedded Linux, MiniGUI/Embedded, Image Processing, GUI Development

---

### Introduction

According to research status, there are already many specialized embedded GUI systems for Linux. MiniGUI is one of the mature GUI systems. It is a cross-platform and Linux Thread Library on SVGALib, multi-window GUI support system. Although do not have the Windows platform MFC, OWL, ATL powerful. But, MiniGUI/Embedded has good cross-platform features, light, high performance, high reliability and configurability [1]. More and more of them are used in the high-end mobile phone, PDA products, and industrial automation control. The traditional image processing system due to lack of support from third-party software development, so they rarely used in the high-end equipment. Moreover, with representatives of the MFC for the Windows platform which developed by image processing technology applied in those applications is much less. Therefore, how to transplant Windows MFC mature image processing technology into the MiniGUI/Embedded becomes the current problems to be urgent solved. The process includes the establishment of the compiling environment, file system design,

modular management, the algorithm analyzes from the source file as example, the solution to the key problem, and debug and run to verify experiment result in Virtual frame buffer.

## Minigui/embedded development platform

### A. The Hardware Platform of MiniGUI/Embedded

Embedded systems take Samsung's S3C2410 processor core, it internal integrated 32-bit ARM920T processor hardware core, the standard of frequency is 203MHz, the hardware platform and extending the 64MB NandFlash, 2MB NorFlash, 64MB SDRAM, and master/slave USB, etc. [2]. S3C2410 development board platform through the debug serial port, JTAG parallel port, Ethernet port communicate with host, image processing interface through LCD display, touch screen/mouse to control.

The hardware platform architecture shown in Figure 1:

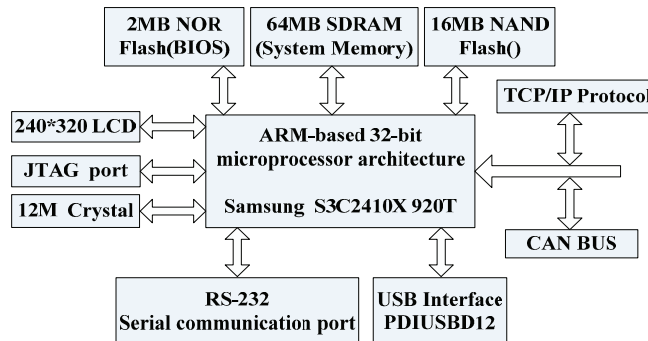


Figure 1. Hardware platform architecture

### B. The Software Platform of MiniGUI/Embedded

Install Linux Redhat9.0 operating system as the host development environment, through the minicom to debug program. Target board transplantation S3C2410 ARM-Linux, and the development board must be equipped with Minigui1.6.10 environment in order to that MiniGUI development of embedded Linux applications can be published to the file system.

## The process of translating image processing technology of windows into minigui/embedded

### C. Configuration of MiniGUI/Embedded Development Environment

First install the compiler armv4l-unknown-linux-gcc. Second, install libminigui-1.6.10.tar.gz, minigui-res-1.6.10.tar.gz, and qvfb-1.1.1.tar.gz, activate Frame buffer and run qvfb mode. The configuration process to select 640 \* 480 "VGA" Depth 32 bit. Pay attention to compile application of MiniGUI in host, to add -lminigui -lpthread -lm -lz -lmgext -lutf -lpng -ljpeg link option, which are used to support graphics library, the thread library, MiniGUI core library, PNG graphics support libraries, etc.

### D. The Design Process for Image Processing System

The main goal of this system is how deal with mature image processing algorithms of Windows implanting into MiniGUI. So that it can realize the Bmp bitmap to display and algorithm processing. The system can open, save, and print, geometric transformation, filtering, and morphological processing the picture. First, create a file system including the Include folder, Obj folder, Src folder, Res folder. Second, need to write three types of files: header file (.h), implementation file (.c) and the main function file (main.c). The entire file system design and build process shown in Figure 2:

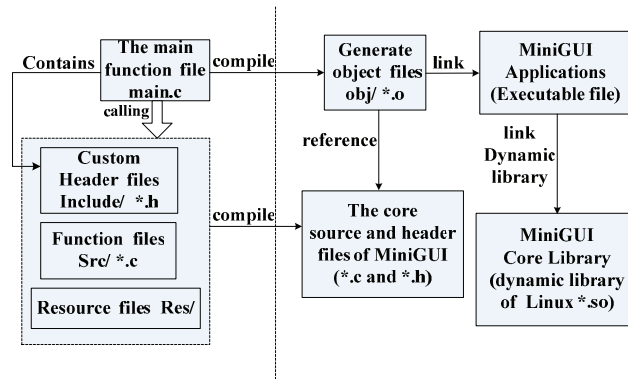


Figure 2. MiniGUI/Embedded file system design and build

### E. Design of Header Files

In view of the embedded system, because of lacking of Windows MFC powerful image definition of the structure and operations support, so the image requirements for the operation about the structure, variables, and most of the algorithms need to be defined by ourselves. The file BMPMAP.h is a bmp header file contains a bitmap structure to be defined, including the bitmap file header BITMAPFILEHEADER, bitmap information header BITMAPINFOHEADER, palette Structure Palette, and the actual bitmap data ImageData [3]. MAINUI.h contains menus, messages, custom control the ID mark, also covers image processing of the macro definitions and function declarations.

### F. Design of the Corresponding Implementation Files

MFC provides a large number of sophisticated image processing algorithms in Windows. In the transplant processing, the system takes modular design based Software Engineering principles of high cohesion and low coupling and each specific algorithm divides into a single module. So the module contains the specific functions of the internal implementation, while minimizing the correlation between the modules. Modules are divided as follows: image preprocessing module, to change a bitmap image format into the data which is processing needs; image processing module, to use Windows sophisticated image processing algorithms on the images of various operations; image storage modules ,to store the processed image and compressed. Module structure of the system shown in Figure 3:

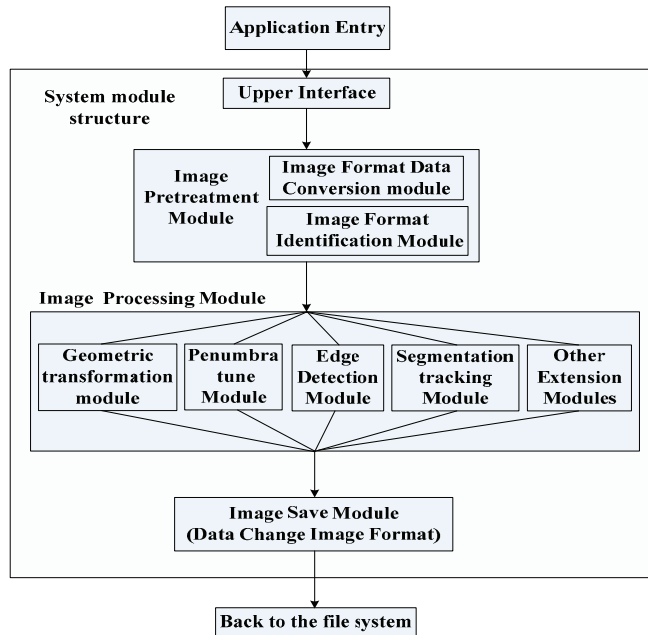


Figure 3. System module structure

The following will combine with the image segmentation algorithm; to explain how the Windows image processing technology is under the transition to MiniGUI/Embedded platform.

First of all, through the image preprocessing module, change the Windows bitmap format into the format corresponding to the treatment.

Secondly, the image processing algorithms under Linux need to conform to the Windows, image processing algorithms. Otherwise, after the treatment the image processing results are quite different. The best image segmentation threshold algorithm [3] is as follows:

- a) Find the minimum and maximum image gray value is  $Z_l$  and  $Z_k$ , the initial threshold is

$$T_0 = \frac{Z_l + Z_k}{2}.$$

- b) According to the threshold  $T^k$  image divided into two parts, the objectives and background Calculate the average gray value of two parts  $Z_O$  and  $Z_B$  :

$$Z_O = \frac{\sum_{z(i,j) < T^k} z(i,j) \times N(i,j)}{\sum_{z(i,j) < T^k} N(i,j)},$$

$$Z_B = \frac{\sum_{z(i,j) > T^k} z(i,j) \times N(i,j)}{\sum_{z(i,j) > T^k} N(i,j)},$$

$Z(i, j)$  is the image  $(i, j)$  point of the weight factor, generally  $N(i, j) = 1.0$ .

c) Find the new threshold:  $T^{k+1} = \frac{Z_O + Z_B}{2}$ .

d) If  $T^k = T^k + 1$ , the end. Else  $K \leftarrow K + 1$ , turn step 2.

Finally, save the module by the images, the processed image compression to be saved.

Throughout the transition process, the Idea of the algorithm must be consistent with MFC. The most of program identical to MFC, but there is the special changes, when use and change image data, need Globallock () global memory management which is API functions to lock the memory in Windows. So that from the specified block of memory to read and write. But in Linux, there is no such as Globallock () API function, so use malloc () function to instead. The global function Globallock () are slower than other memory management functions and do not provide as many features. However, the global functions are still used with DDE and the clipboard functions [4]. Thus, as a platform to MiniGUI speed image processing system is faster than under Windows.

### G. Design of the main Function

The main function is to set the main window, customize menus and toolbars, and manage the whole program. It mainly to create and initialize window and set the coordinates of the image display's location and size. MiniGUI event based message-driven mechanism to trigger the application of the event into a message. Application of these events through the window function associated window procedure to handle [5]. The main function of the event message handles process in general as shown in Figure 4:

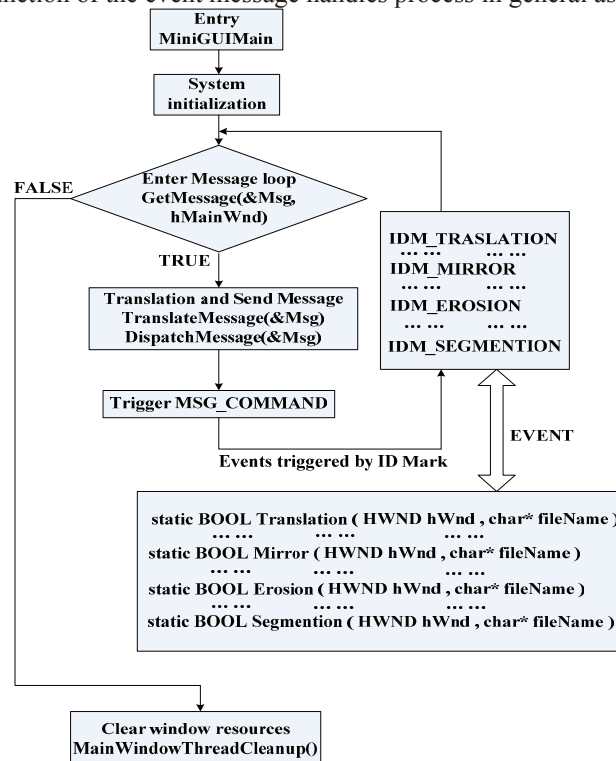


Figure 4. The main function of information processing in the event

#### H. Image Processing System Interface Run on Virtual Frame Buffer

To run executable file in Virtual frame buffer of Linux to verify program, so that changes can be avoided in the development and debug board to bring the red tape. So far, the frame buffer can be displayed on PC machine main interface, and in this basis, repeatedly modify the source code to debug the image processing system in order to achieve the same effect under Windows environment. So that the whole system to achieve the desired effect. The system interactive interfaces Figure 5:

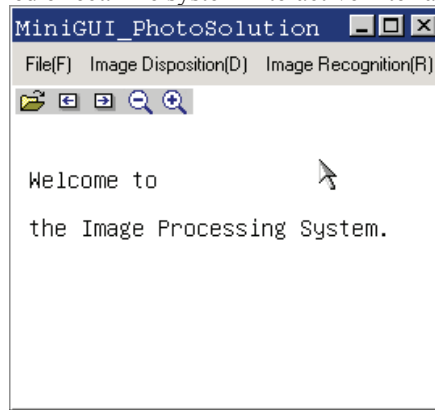


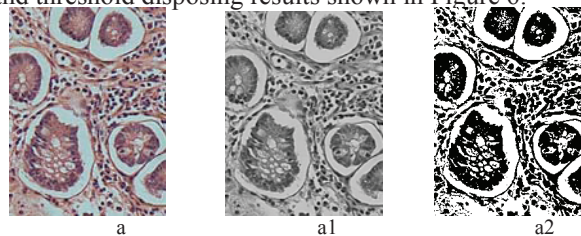
Figure 5. User interface

#### I. Make Static or Dynamic Link Library

In process of image, there are MFC and API to support in windows which improved development efficiency and quality. Learn the technology product some common functions into a Linux library (static library and dynamic library). Static library is compiled in the program will be linked to the target code, while the program is running is no longer needed library files. Dynamic library is the program compiled and will not be linked to the target code, but when it is running the library file is loaded. Make BMPMAP.h and bmpOperation.c documents into static and dynamic library file library to improve the efficiency of the system.

#### Experimental results

In Windows, run the Image algorithm program. To perform MiniGUI executable file at the Virtual frame buffer. The original image with a load, in both circumstances, after processing, through debugging, operating results in the image data formats are consistent. Further, the picture was same. The original image was through gray and threshold disposing results shown in Figure 6:



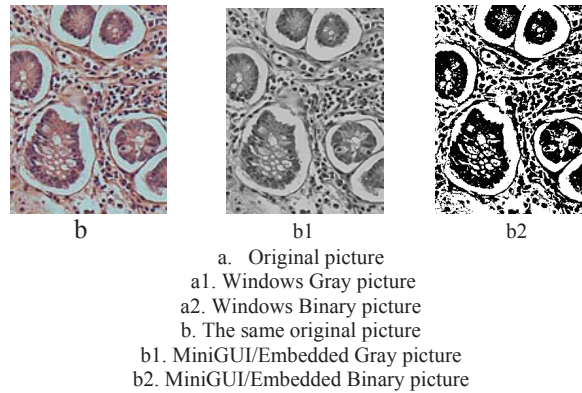


Figure 6. Windows and MiniGUI/Embedded results contrast

## Conclusion

By experiments, it is show that the system is user-friendly, low-cost, lightweight, fast processing speed, portability and strong features. The system can be customized as needed to improve medical image analysis for embedded, mobile photo processing, PDA picture browsing and handling, satellite tracking and other image capture applications.

## References

- [1] Beijing Feynman Software Technology Co.Ltd. MiniGUI Tech White Paper-1.3C Beijing: Beijing Feynman Software Technology Co., Ltd.2003: 3-6.
- [2] Samsung Electronics S3C2410A – 200MHz & 266MHz 32-Bit RISC Microprocessor User's Manual, Revision 1.0 (March 2004) Yon gin-City : Samsung Electronics 2004:31-35.
- [3] Bin He, Tianyu Ma, etc. ed. Visual C ++ Digital Image Processing (Second Edition).Beijing: Posts & Telecom Press, 2002.
- [4] Jeffrey Richter 《PROGRAMMING APPLICATIONS FOR MICROSOFT WINDOWS FOURTH EDITION》 Fourth Edition Publication Date: 2005.
- [5] Ligong Zhou. ARM Embedded Application Development and preliminary examples MiniGUI [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2006:113-117.