Open Source OCR Framework Using Mobile Devices

Steven Zhiying Zhou*, Syed Omer Gilani and Stefan Winkler

Interactive Multimedia Lab, Department of Electrical and Computer Engineering
National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576

ABSTRACT

Mobile phones have evolved from passive one-to-one communication device to powerful handheld computing device. Today most new mobile phones are capable of capturing images, recording video, and browsing internet and do much more. Exciting new social applications are emerging on mobile landscape, like, business card readers, sing detectors and translators. These applications help people quickly gather the information in digital format and interpret them without the need of carrying laptops or tablet PCs. However with all these advancements we find very few open source software available for mobile phones. For instance currently there are many open source OCR engines for desktop platform but, to our knowledge, none are available on mobile platform. Keeping this in perspective we propose a complete text detection and recognition system with speech synthesis ability, using existing desktop technology. In this work we developed a complete OCR framework with subsystems from open source desktop community. This includes a popular open source OCR engine named Tesseract for text detection & recognition and Flite speech synthesis module, for adding text-to-speech ability.

Keywords: Mobile devices, OCR, Text-to-Speech, Open source

1. INTRODUCTION

Traditionally mobile phones were used as a convenient wireless voice communication tool. However with availability of more computing power and memory on mobile platform this is no longer the case. Mobile phones have evolved from passive one-to-one communication device to powerful handheld computing device. Today mobile phones can capture images, record videos, manage organizers, keep journals, browse internet and perform much more complicated tasks. Mobile phone vendors, operators, OEMs and ISVs are shipping innovative applications with new installments.

This progress identifies the rapid maturity of mobile platform in terms of running complicated applications. Interesting consumer application are emerging on the mobile landscape like smart tour guides, business card readers, autonomous navigation, document compression etc., The quality of service these applications provide is very much dependent on the image captured with the integrated camera. In most cases images are noisy due variation in brightness, contrast and illumination levels. Furthermore images captured with the integrated camera usually suffer from motion blur and perspective distortion problem as well.

Image noise is not the only problem we encounter on mobile platform. Software running on mobile platform is still constrained by certain limitation, like lack of floating point units (FPU) and small visual displays. Moreover for running complicated algorithms, like used in image processing subsystem of OCR, we need more computational power and more memory space.

However with all these exceptions and constraints mobile platform is evolving and maturing rapidly towards providing better services and application for the consumers.

In present work we investigate the deploying of a framework on a mobile platform with components that were not originally designed and developed for it. We used opens source software components from the traditional desktop platform and developed a complete OCR framework for mobile platform. Furthermore to make the interface more

*Corresponding Author: elezzy@nus.edu.sg

intuitive to user and to provide a complete framework we added a text-to-speech (TTS) engine for audible results. TTS converts the OCR recognition results in to audible speech output.

The rest of the paper is organized in to following sections. Section 2 gives previous work. Section 3 shows system specification and design constraints Section 4 illustrate the over-all approach. Section 5 discusses core framework and explains the functionality of each sub-system of the core. Section 5 describes the results and Section 6 envisions future directions.

2. PREVIOUS WORK

Today most of the mobile phones are equipped with good resolution digital cameras. This enables the mobile phones to run image based application that in past require external digital cameras.[1] However there are still some limitations to what we can do on mobile platform. In Context of running an OCR application these limitation pose interesting challenges for research and development. Mobile phones have limited power to run complex software, like OCR engines, using their own limited hardware and memory resources. Huge amount of pixels are to be processed in limited amount of on-board memory compared to desktop systems with GB of memory and additional virtual memory mechanism. Not to mention the more high resolution the images are the more computationally taxing task it is to run the OCR on mobile phones. Another shortcoming is the absence of the hardware support for real number processing on the mobile platform. Real number operations are simulated using float emulation libraries which affects the efficiency of the software itself. In addition small displays panels make it a difficult task to design a good user interface. [2]

The first prototypes of mobile based OCR appeared in 2001 and commercial-level products for consumers were launched in 2002.[3,4] OCR is complex tasks including rigorous image processing and complicated algorithm thus heavily taxing its CPU.[5] Usually the image captured with mobile phones suffers with different problems like motion blur, perspective distortion, variations in illumination and contrast. Furthermore the scene text varies in size, fonts and orientation. Text image undergoes several levels of processing before the final result is presented to user.[6]

Many commercial-level applications are now available in market for OCR on mobile devices. Hitachi has developed an OCR with 1MB memory footprint.[5] However it uses external server to perform computationally expensive task of OCR. Pantech-Curitel offers OCR functionality using JPEG images.[5] Xerox has developed dedicated image processing technologies for OCR on desktop platform.[7] ABBYY has developed a professional level cross-platform OCR SDK for the developers to create small footprint OCR solutions for mobile platform.[8] However the technology is commercial and is not freely available.

3. SYSTEM DESIGN AND SPECIFICATIONS

We used Hp iPAQ model rw6828. It comes with a 416 MHz ARMV4I processor, 128 MB Flash ROM and 64 MB of RAM shared by memory and code space. It can take digital photo up to 1600x1200 resolutions under adjustable luminance, contrast and saturation. It uses 2 mega pixel integrated camera. It is configured with Windows Mobile 5.0 Operating system and has a 240x320 TFT display panel in a portrait view.

A complete framework is designed in a modular fashion such that any one module can be replaced with a better alternative without affecting other modules of the framework. A system diagram of developed prototype is shown in Figure 1

As shown in the system diagram prototype comprises of hardware and software components.

3.1 Software Component

It includes two sub components, framework specific core components and pre-installed operating system. Framework is built with a simple GUI. DAI stands for the Digital audio interface and used for output of text-to-speech. Adapter is used to transform the input data (compressed digital image) to output data, readable by OCR engine.

Essentially the application core components like Adapter, OCR engine and TTS (text-to-speech) synthesizer are available for the PC platform under open source umbrella. They were ported to mobile platform using Visual Studio 2005 IDE and Window Mobile 5.0 pocket PC SDK. The ported components include DJpeg Library, Tesseract OCR Engine [9] and Flite Voice Synthesizer.[10]

3.2 Hardware Components

It includes integrated camera to capture digital image, display panel for GUI and speakers for the output of speech synthesis.

4. METHODOLOGY

Using the existing open source modules we developed a complete Text-to-Speech System. A schematic representation of the software process is shown in the Figure 2. A Digital Image is captured using built-in camera of iPAQ. The captured image is saved at resolution of 640x480. Although we had access to higher resolution but we found that 640x480 gives good results with reasonable time for text processing. The VGA image is then converted to 1-channel MS-Windows bitmap format. The intensity bitmap is used by the OCR engine. Selected OCR engine has text detection and recognition capability. It translates the input images to ASCII text files. Once the recording of OCR results on text files is completed we proceed to post-processing stage. In this stage we remove all the non alphabetical and numerical characters from the script. This process can be further refined by doing a spell check to identify possible misclassification of the characters at word-level or word at phrase-level. Based on the classification we correct or discard the word. After this post-processing the text file is loaded by Text-to-Speech engine. The engine synthesizes speech in real time, at 8 KHz. Finally audible output is pushed to built-in speakers of iPAQ.

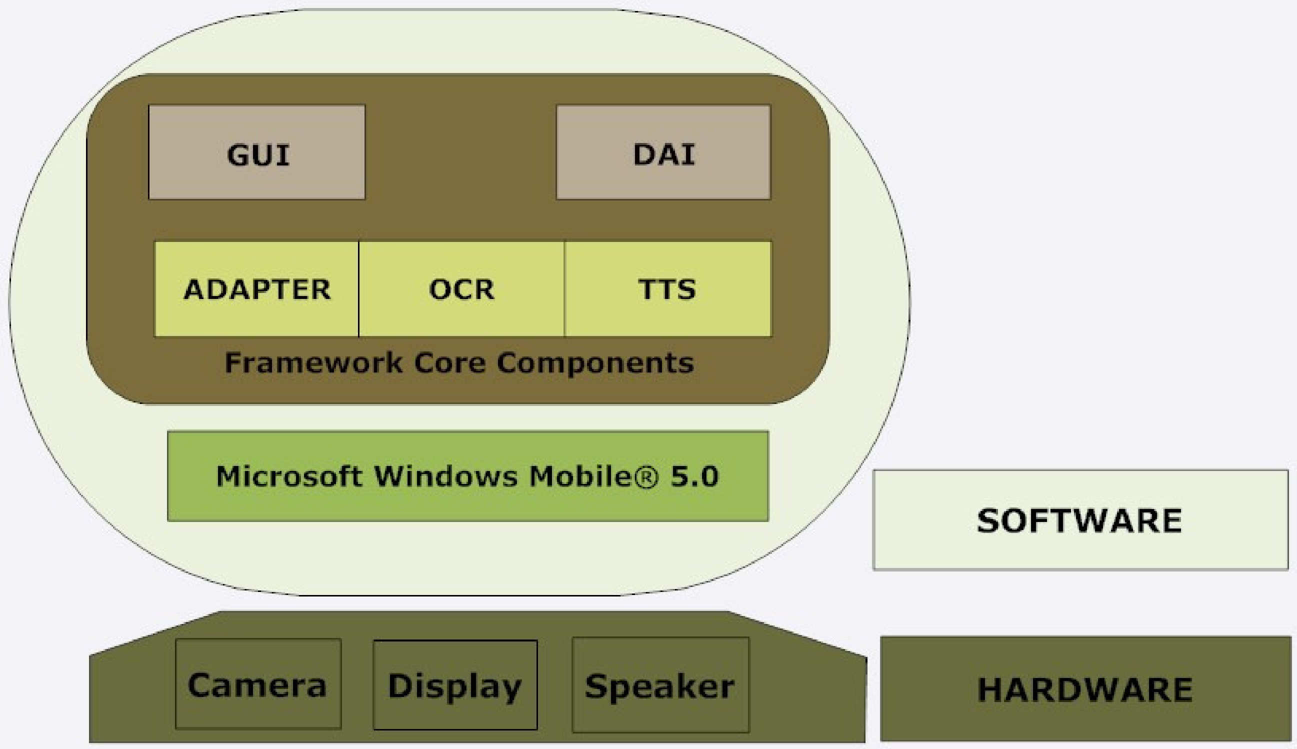

Figure 1 System diagram of the prototype

5. CORE FRAMEWORK

5.1 Image-Preprocessing

iPAQ has a 2 Mega Pixel, CMOS matrix, built-in camera with high quality F/2.8 lens. Its focus is set to 1.8 M. We can capture images up to 1600x1200 resolutions under varied luminance, saturation, sharpness and contrast settings. Initial images are in compressed Jpeg format. We need an adapter to convert these compressed jpeg images to bitmap images.

We use freely available DJpeg library for the image format conversion. The images after conversion are still in RGB format. In order to reduce the processing time taken by OCR we convert the multi-channel RGB image into single-channel intensity image.

5.2 Text Detection and Recognition

In developed prototype system text detection and recognition is performed by a single module. This module is implemented by using a freely available, open source OCR engine. We selected Tesseract[9] for this purpose.

Tesseract is a quality open source OCR engine developed at HP, between 1985 and 1995. It was among the top-tier performers at OCR competition organized by UNVL. In 2005 it was handed over to Information Science Research Institute (ISRI) which later on, with Google Inc, made it publically available as open source software.

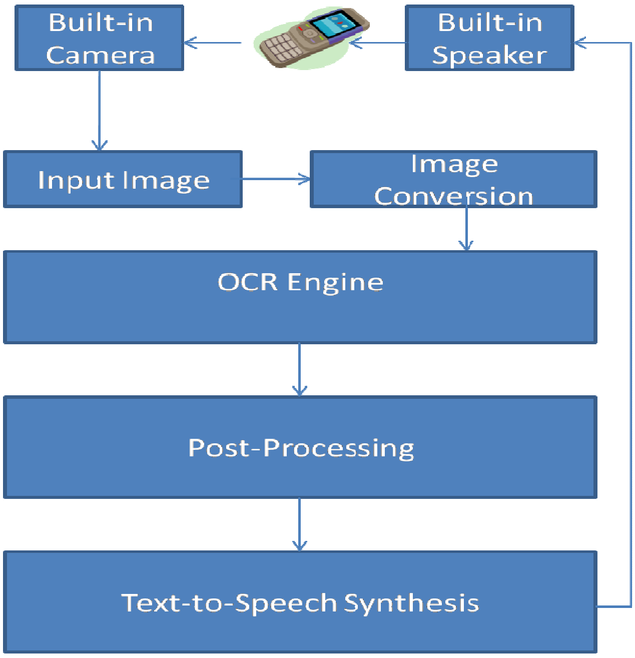

Figure 2 System-level process flow diagram

There are few other OCR engines available in open source landscape as well, like for example OCRAD and GOCR. We tested all three of the OCR engines with our image dataset. We found Tesseract to be leading in results. Figure 3 shows randomly sampled images from our own compiled dataset.

Output result of Tesseract OCR engine is in form of a text file with ASCII coded characters. OCR engine also records a log file which can be useful for debugging purposes. We need to perform some post-processing operation on a ASCII text file before we can pass it to speech synthesizer. This is done by removing all the punctuation marks and symbols other than the alphabets and numeric.

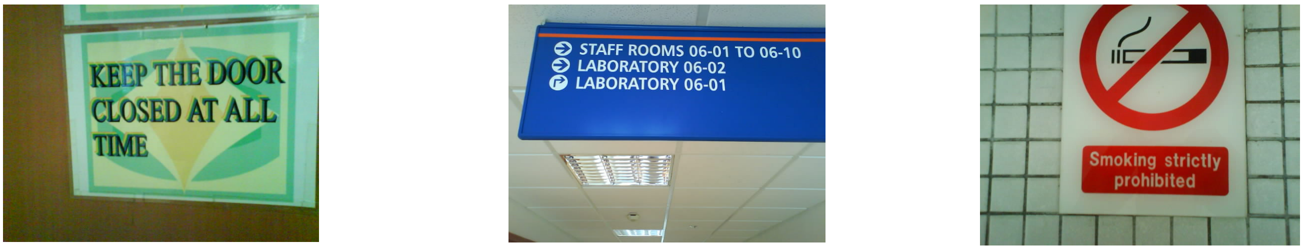
KEEP THE DOOR CLOSED AT ALL TIME
STAFF ROOMS 06-01 TO 06-10
LABORATORY 06-02
LABORATORY 06-01
Smoking strictly prohibited

Figure 3 Randomly sampled images

5.3 Post-Processing

Though we haven't completely implemented this step in our current prototype but it's an important intermediated step between OCR results and final speech output. The quality of the final results (in audible form) very much depends on quality of the OCR results and the post-processing stage.

Post-processing stage basically attempts to remove the noise in output text data by OCR. There can be certain types of noise like non-alphabetic or non –numeric characters, group of characters making no meaningful word etc.

To eliminate the non-alphanumeric character noise we can simply filter out OCR results based on ASCII character code limits. To make the framework more robust the filtered text can be further processed by cross referencing it with the dictionary and using high level semantics. This would correct any spelling mistakes, in OCR results, and help in removing alphabets which do not make any meaningful word.

5.4 Speech Synthesis

We used Flite speech synthesis engine for our text-to-speech module. Flite [10] is small and fast run-time speech synthesis engine developed at CMU. It's an alternative of Festival engine. Festival engine was not suitable for the small scale systems. To address the problem of scalability on devices with small memory and power it was necessary to develop a light version of the speech synthesis engine. Flite TTS engine also embodies many improvements over its predecessor like fixed point arithmetic, portability and smaller synthesizer.

Filtered OCR results are forwarded to Flite TTS engine for speech synthesis. Default voice distributed with the Flite engine is of 8 kHz. This makes the final audible output more machines like than human like.

6. RESULTS

In this paper we described a complete OCR framework on a mobile phone. The proposed framework uses existing open source desktop technology.

We tested the core framework with different resolution images. We found that images with 640x480 gave the best results, in terms of processing speed and accuracy. The average time for the whole system, from image capture to final speech output was approximately 8 to 12 seconds.

The developed framework can be used in variety of scenarios. For example the framework can be used as basis for developing the business card reader application, a tourist guide, a road sign detector and translators etc.,.

7. FUTURE DIRECTIONS

We have developed a complete text recognition framework, with TTS capability, using existing open source desktop technology. The motivation here was to provide a general purpose framework suitable enough to develop more exciting application over it. Moreover since the framework is developed in a modular fashion it is very easy to make updates on core components. We can update any single module without affecting other modules in the core system. we can for example replace the existing OCR engine with a better and more robust but open source OCR solution.

Future work includes, refining the results by OCR using cross-referencing with the dictionary and other high level semantic techniques, developing applications using the core framework and making modification to existing OCR solution for more robust results. We also plan to complete a tourist guide application and conduct the user study.

REFERENCES

[1] Ezaki, N., Kiyota, K., Minh, B. T., Bulacu, M., and Schomaker, L. “Improved Text-Detection Methods for a Camera-based Text Reading System for Blind Persons”. *In Proceedings of the Eighth international Conference on Document Analysis and Recognition* (August 31 - September 01, 2005). ICDAR. IEEE Computer Society, Washington, DC, 257-261.

[2] Jing Zhang, Xilin Chen, Jie Yang, Alex Waibel, “A PDA-Based Sign Translator”. *In Proceedings of the 4th IEEE international Conference on Multimodal interfaces* (October 14 - 16, 2002). International Conference on Multimodal Interfaces. IEEE Computer Society, Washington, DC, 217.

[3] “Multimedia Mobile Appliances with OCR”, *Axis*, Vol.92, P.62, Tokyo, 2001.

[4] S.Senda et al. “Camera –typing interface for ubiquitous information service”, *Proc. 2nd International Conference on Pervasive Computing and Communications*, Florida, USA, 2004

[5] Mikael Laine; Olli S. Nevalainen, "A Standalone OCR System for Mobile Camera phones," *Personal, Indoor and Mobile Radio Communications,* 2006 IEEE 17th International Symposium, vol., no., pp.1-5, Sept. 2006.

[6] Bae, K.S.; Kim, K.K.; Chung, Y.G.; Yu, W.P., "Character recognition system for cellular phone with camera," *Computer Software and Applications Conference*, 2005. COMPSAC 2005. 29th Annual International , vol.1, no., pp. 539-544 Vol. 2, 26-28 July 2005

[7] Newman, C. Dance, A Taylor, S. Taylor, M. Taylor, T. Aldhous, “CamWorks: A video-based tool for efficient capture from paper source documents”, *Proceedings of ICMCS*, June. 999, pp.647-653

[8] ABBY Mobile OCR SDK 2.0, http://www.abbyy.com/sdk/?param=56223

[9] http://www.code.google.com/p/tesseract-ocr/

[10] Black, A. and Lenzo, K. (2001) Flite: a small fast run-time synthesis engine, pp 157-162, ISCA, *4th Speech Synthesis Workshop*, Scotland.