Mobile IC Package Recognition and Information Retrieval

Chris Young
cjyoung@stanford.edu

Patrick Blaes
prblaes@stanford.edu

Overview

Modern electronic systems are typically built using many integrated circuits (ICs) which are soldered to a printed circuit board. These ICs are usually contained in small, rectangular packages with a cryptic looking part number printed on them. When examining an unfamiliar circuit, identifying each IC and obtaining information about it can be a tedious procedure. This usually involves locating the part number on the chip, typing it into a part supplier website such as Digikey, and sifting through search results to find the part's datasheet.

We propose a new image processing system which will make IC part identification and information discovery significantly faster and easier. Our system will allow the user to take a photograph of a circuit board using their Android smart phone. An example circuit board photograph taken with a Motorola Droid is shown in Figure 1. The ICs will then be automatically identified, and part names and datasheets will be presented to the user in an easy-to-use touch interface.

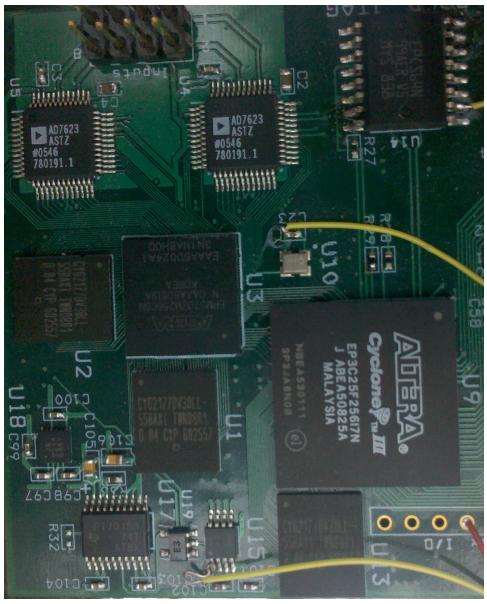

Figure 1: Example input image.

Implementation

Once an image of a PCB is acquired, the processing pipeline is as follows:

1. First, each chip is localized based on its rectangular shape possibly using the Hough transform [4] or thresholding based on the dark gray color of each chip compared to the usual green solder mask of the PCB.

2. Each chip is extracted from the original image.

3. We then perform contrast enhancement on the extracted chip image as the text on each package is usually white (which is already contrasty) or yellow (which is more difficult to see). The contrast enhancement could be accomplished with histogram equalization.

4. The algorithm then localizes the position of the text on each chip [1], [3], [5].

5. Following localization, we then segment the text for each chip image.

6. The text image is then sent to a server running an OCR engine, possibly Google's open source Tesseract OCR [2].

7. Once the text is identified, the server sends the part numbers to the part supplier Digikey and acquires the data sheet and sends it along with the chip name to the phone.

8. The UI will make use of the Android touch interface for selecting chips and viewing datasheets by drawing an outline around each chip.

We will have to experiment a little to decide how much of the pipeline is implemented on the server vs. the phone itself. It may be that we can perform the whole pipeline on the phone by making use of a cloud based OCR engine in addition to including connectivity to the users PC to open datasheets. Ideally, we would also be able to perform the whole operation in real time making use of the live camera mode. However camera resolution constraints and complexity of the algorithm may limit how much processing the phone can accomplish in a reasonable amount of time.

At the very least, we will complete the project using still images and perform processing on a server, and time permitting, we can experiment with the UI and the above mentioned possibilities.

References

[1] B Epshtein, E Ofek, and Y Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2963–2970, 2010.

[2] Google. Tesseract OCR Engine: http://code.google.com/p/tesseract-ocr/.

[3] Y.M.Y Hasan and L.J Karam. Morphological text extraction from images. *Image Processing, IEEE Transactions on*, 9(11):1978–1983, 2000.

[4] C.R Jung and R Schramm. Rectangle detection based on a windowed Hough transform. In *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, pages 113–120, 2004.

[5] Huiping Li, D Doermann, and O Kia. Automatic text detection and tracking in digital video. *Image Processing, IEEE Transactions on*, 9(1):147–156, 2000.