TJHSST Senior Research Project
Devlopment of a Generic Font OCR
Final Project Proposal
2007-2008

Nathan Harmata

November 2, 2007

Abstract

OCR (Optical Character Recognition) is a very practical field of Computer Science. Since the late 1980's, researchers have been developing systems to identify text from non electronic text sources, like pictures or papers. The use of OCR systems has spanned from making books in Braille to sorting mail by zip code by United States Post Office.

Keywords: OCR

1 Introduction

The goal of this project is to create an application that can read text from electronic picture files. One of the main focuses will be developing a generic way to recognize different fonts, rather than hardcoding in definitions for specific fonts. Although OCR is by no means a "new" field, it has still yet to be fully explored. There are very few OCR applications readily available to the public, and even the ones that are free of charge are lacking in performance and consistancy.

For now, an application that can read and recognize "simple" pictures, i.e. ones with minimal headers and only text, will be developed. If that is successful, more advanced techniques to handle headers and "background noise" will be used. This project might also explore the field of handwriting recognition.

2 Background

OCR systems have been around since the late 1980's. Still, they are not widely available or used by the public. The results from a review of the free ones on the Linux operating system are not very promising. Although most of them had measured accuracies above 94 percent, that is not good enough. The one commercial product tested, Aspire OCR, was accurate only 91.5 percent of the time. The most likely industry standard, Tesseract, is also one of the oldest OCR systems. The review measured it to have an accuracy rate of 99 percent. Development on it started in 1985 and it is still used as the OCR engine for Ocropus, Google's textual analysis application. It is unlikely that this project will be able to achieve similar success, but the goal is to be on par with the current OCR options.

3 Procedures

The application will accept a picture file as input. It will then break apart the text in the picture into headers, and paragraphs. Paragraphs will be broken down into sentences, which in turn will be parsed into words and spaces. The idea of this is to then have individual pictures of each letter, which can be interpreted as a graph of pixels. Each character will be analyzed so that the best match from a database can be determined. The purpose of the database is to contain general definitions of the letters of the alphabet, so that many different fonts can be recognized. For now, pixel counts for the four quadrants of each letter, and slope fields will be used for this comparison. Eventually more advanced methods will be incorporated.One of the more simple improvements is to analyze each letter in the context of its word and in the context of a sentence. That way, the database doesn't have to return the absolute best match but only the best matches. Using a dictionary and/or a grammar reference, a very accurate determination of each letter can be made.

3.1 Testing

1. A user-friendly interface for viewing the pictures of parsed words has already been developed. This allows for manual detection of errors in letter parsing.

2. Once more work is done on developing the database of general letter definitions, as described above, a more automated and effective method of testing will have to be employ. One idea is to write a script that creates picture inputs from randomly generated combinations of letters and then compares the results from the OCR application with the actual text. Another idea is to develop a way to test the accuracy of the OCR system in different areas, like how well it performs on individual letters versus complete sentences.

3. The current method of making direct pixel comparisons has encountered several problems. First off, in the Courier font, certain capital letters "elide" together. By this, it is meant that there is no vertical space between adjacent letters. One example is the combination of "A" and "B" to form "AB." Thus, the program interprets "AB" as a character that is the combination of "A" and "B," which of course is not going to yield a direct match with the cache. An algorithm to discern between elisions and actual characters will have to be developed.

The following computer languages, algorithms and programs will be used.

3.2 Software

1. Java will be used for picture input and output.

2. C may end up being used for the complex vector operations needed to compare letters to the general database.

3.3 Algorithms/Programs

1. KolourPaint will be used to make picture files for input and to precisely view pictures.

4 Expected Results

It is expected for the final product to at least be able to recognize the text of a simple picture with very high accuracy if the used font is a common one, such as "Times-New Roman" or "Courier." The final product should also be able to effectively deal with less common fonts using generic letter definitons. These results can easily be presented by explaining how the program works - how it parses a document and how it analyzes the text. Development of a way to visually show the computations being done may be considered to aid in this.