Automatic Equation Solver and Graph Plotter

EE 368 Final Project, Spring 2012

Aman Sikka (asikka@stanford.edu), Benny Wu (bennywu@stanford.edu)

Motivation

The rapid growths of computing power and access over the last several years enabled a move away from rigid lecture based models where the entire class is instructed at the same pace to interactive learning models where students work at their own pace. One such example is the Khan Academy's pilot project with the Los Altos school district. The drawback with the current technology is the limited feedback, often consisting of only whether a question was answered correctly or incorrectly. Our goal is to develop a program that bridges the gap between technology and the traditional pen and paper approach – in essence a content and context aware smart notepad. For this class we will simplify the problem to an automated equation solver. We will propose the full version of the design for the 'Magic Grant'.

Goal

Given an image of a set of equations or a function to be graphed, generate a step-by-step solution or a plot.

Approach

The general approach is to detect text in an image to identify both keywords (such as "graph" and "solve") and equations. The equations can then be passed to a solver that will show the solution in a step-by-step manner. The pipeline can be broken down into four sections as shown in Figure 1.

1. Pre-processing: binarization of camera image using locally adaptive Otsu's method, noise reduction and skew correction similar to as described in [2].
2. Segmentation: isolate individual characters.
3. Classification: meaning from the text will be extracted using the Tesserac OCR engine [3].
4. Post-processing: the commands will be passed to a solver. Binary expression tree will be used to parse the expression and solve the equation [4]. A step-by-step solution and graph (when applicable) will be shown on screen.

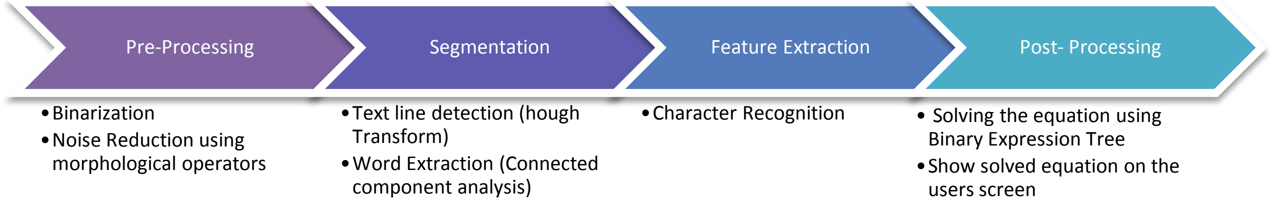
Pre-Processing
Segmentation
Feature Extraction
Post- Processing
• Binarization
• Noise Reduction using morphological operators
• Text line detection (hough Transform)
• Word Extraction (Connected component analysis)
• Character Recognition
• Solving the equation using Binary Expression Tree
• Show solved equation on the users screen

Figure 1: Program Flow

Initially, we will assume that the image contains only the equation to be solved or graphed in a computer font. If time permits, we will relax the constraint to include images of hand-written printed characters. We will be using the Android phone as a development platform for this project

References

[1] Jian Liang, David Doermann, Huiping Li. "Camera-based Analysis of Text and Documents: A Survey", Int. J. on Document Analysis and Recognition, 2005.

[2] Sam Tsai, Huizhong Chen, David Chen, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod, "Mobile Visual Search Using Image and Text Features", Proc. 45th Annual Asilomar Conference on Signals, Systems and Computers, 2011.

[3] Tesseract OCR engine: http://code.google.com/p/tesseract-ocr/

[4] Binary Expression Trees: http://www.cs.utexas.edu/~lavender/courses/EE360C/lectures/lecture-24.pdf