Sudoku Solver

Yixin Wang
wangyix@stanford.edu

Sudoku puzzles are a common sight in newspapers and puzzle books, and sometimes they don't come with solutions. While it's possible to manually type in the given numbers into a solver online, it would be much faster and more convenient to solve this problem through image processing.

The goal of this project is to create an Android app that can automatically solve a Sudoku puzzle from images taken by the phone camera, either in real-time or close to real-time. Ideally, the user would simply need to aim the camera at the Sudoku puzzle, and the app will automatically detect the puzzle, solve it, and overlay the solution on top of the camera image.

The steps of the algorithm may look something like the following. First, the image is binarized using some method of locally adaptive thresholding for documents, such as in [1]. Then, noise will be reduced using morphological opening. The orientation of the Sudoku puzzle will be estimated using the Hough transform. The existence of 10 vertical lines and 10 horizontal lines will help verify that this is indeed a Sudoku puzzle. The Sudoku puzzle will then be perspective-corrected so that its grid lines are axis-aligned. The given digits in the grid cells will be extracted and recognized using template matching, or possibly with a more complex method that will also handle hand-written digits, such as [2]. Once the digits have been recognized, the puzzle will be solved using some Sudoku solving algorithm. One such method is to formulate the Sudoku problem as an exact cover problem and to solve that using dancing links, outlined in [3]. Finally, the results will be overlaid onto the screen image by adding perspective-transformed images of digits into the empty grid cells of the puzzle. A possible extension to this app could be the ability to give the user a hint on a half-solved puzzle by revealing just one digit somewhere that can be deduced based on the current state of the puzzle.

I will not be using a Droid phone for this project; I will use my own Android phone.

References

[1] A. E. Savakis, "Adaptive Document Image Thresholding Using Foreground and Background Clustering", *Image Processing*, vol. 3, 1998, pp. 785-789, Oct 1998.

[2] A. K. Jain, "Representation and Recognition of Handwritten Digits Using Deformable Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, issue 12, Dec 1997.

[3] D. E. Knuth, "Dancing Links", *Millennial Perspectives in Compute Science,* pp. 187-214, 2000.