Klokan - Program Documentation

Jan Horešovský

April 2017

Klokan is a program which can be used to evaluate answer sheets from the Mathematical Kangaroo.

1 User Documentation

There will be a thorough manual on how to use the program here once user interface is ready.

For now:

- After startup Klokan will prompt you to enter the path to a .jpg file with a scan contating the correct answers.
- Then you will enter paths to other scans in a loop. These scans will be evaluated and the final result outputted.
- When all the answer sheets for the current instance are scanned, you can choose either to begin a new instance with a new set of correct answers or to quit the program.

2 Developer Documentation

This section describes the architecture of Klokan and contains a thorough description of all interface functions that can be called from outside of the program. At the end there is a subsection dedicated to certain technical details of other functions that are worth mentioning.

2.1 Overview

The program is based on deterministic image recognition using simple geometric algorithms. These, and also some other functionality necessary for image pro-

cessing is taken from the OpenCV library version 3.1.0. The whole program is written in C++.

The process of evaluating an answer sheet comprises three stages:

- Table Extraction
- Cell Extraction
- Cell Evaluation

These also correspond to the three interface functions that this program provides.

2.1.1 Table Extraction

In order to extract tables from the answer sheet the program needs to find them first. Tables are generally the largest object in the sheet, so instead of looking for tables the program simply looks for the largest object. This is done by using an OpenCV function `flood_fill()`. This function takes a pixel and floods it and all its neighbouring pixels (and all their neighbouting pixels...) which have the same colour as the starting pixel. Then it returns a number which represents the area which has been flooded. This way the program finds out where the upper left pixel of the largest object is. When this table is extracted, it is hidden (flooded with black) and the second largest object is found.

After the program finds a table, it creates a copy of the sheet in which it hides all other parts of the image. Then it uses an OpenCV function called `hough_lines()` to find all the lines in the table. The function is set so that it only finds lines that are at least as long as the expected edge of the table (this is deduced from the size of the input image). After that the corners of the table are found by taking the extreme lines and their intersections.

The table does not generally have to be perfectly aligned with the edges of the image. However, this is taken care of by using the corner points and an OpenCV function called `crop_perspective()` which maps these corner points onto the corner points of a square. This square has a side equal to the edge of the table. After that other points are transformed accordingly and the program obtains a new image containing only the table which is now perfectly aligned. This step is important for being able to cut the table into cells easily.

2.1.2 Cell Extraction

2.1.3 Cell Evaluation

2.2 Interface Functions

Doxygen output is planned to be here.

2.3 Other Functions

There will be a description of certain implementation details which are worth mentioning.