Character recognition on game cards

August von Hacht
Chalmers university of technology
Göteborg, Sweden
hacht@student.chalmers.se

David Hansson
Chalmers university of technology
Göteborg, Sweden
davhans@student.chalmers.se

Abstract

1 Introduction

2 Data

The data acquirement will consist of fetching images from online libraries or taking high resolution pictures of real cards. There are plenty of libraries on the Internet containing several images of cards, to note one is for example *http://gatherer.wizards.com* where one can randomly generate cards. These cards can be saved in JPEG format and later loaded into Matlab using the imread function.

In the case where the libraries may lack certain cards containing characters of a specific type or font we also have real cards available. Using a high-resolution camera, pictures of these cards could almost equal well function as data for the project.

3 Methods

The acquired data will be used to build a database, a *classifier*, which will be used in connection with a *feature extractor* to make an automatic procedure for recognising the title of real cards photographed with a web camera. The procedure is similar to an *optical character recognizer* but has a more specific application.

For simplicity, we will, as a beginning, assume that the cards are placed in front of the camera in such a way that the title is clearly visible and the surface of the card is close to perpendicular to the camera. It should also be overall in focus and the card should be inside the camera vision when an image is taken.

Before we setup the procedure, both the feature extractor and the classifier need to be optimized in order to obtain a good performance of the recognizer. This will be done via noise removal and finding a good image size of the database images for optimal correlation measurements. Thereafter, some edge detection filter will be needed to estimate the position of the word and a letter separator in order to compare with the database. The approach will more briefly discussed in the following subsection.

3.1 Noise removal

The web camera is equipped with noise introducing factors which needs to be accounted for. The more critical noise factor introduced is due to that a constant threshold for conversion from gray scale to binary will be used, this due to small variation color levels around the area of interest however some variation exists and has to be accounted for.

Two ways will be tested in order to resolve the noise, the first one is morphological operations and the other is to use a Gaussian filter (Details are to be introduced and discussed for the final report).

3.1.1 Morphological operations

A conversion from gray scale to binary image will allow for opening and closing operations to restore pixels that by noise ended up on the wrong side of the threshold. These will be more throughout introduced here using Matthias Andersson master thesis as a source.

3.1.2 Gaussian filter

A Gaussian filter with a rather hand picked variance will be used in order to reduce the noise around the letters and words in the picture.

3.2 Finding a good image size for database correlation matching

Initially, when setting up the database, the image sizes of the letter was rather small, about 32x16 pixels, which might require very good correlation in order for good matching with the database. But, due to noise we want to see if these images needed to be resize by a constant multiple in order to allow for more noisy camera images. Here the option to resize the database image and keep the camera letter size, resize the camera letter and keep the database image size or both can be used. This needs to be further evaluated in order to find a good method. Similar to forward or backward selection will be used to estimate the method.

3.2.1 Binary image resampling

The resizing of the images will be done via a binary image resampling method with allows for both extended image size and compressed image size, this via binary sequence conservation in the different image sizes.

3.3 Edge filtering and position estimation

A vertical and horizontal edge filter, linear or non-linear, will be used in order to detect the edges from which the title should be within a constant distance of.

3.4 Letter separation and classification

Finally, we need to separate the letters in the words which hopefully is possible using the small white spaces between the letters due to the font used in the cards. The letters will be separated into an array and compared one by one with the database and a the most likely letter will be returned and used.

4 Results

5 Discussion

References

A Appendix