Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-15 (2016) 100–107

Histograms of Stroke Widths for Multi-script Text Detection and Verification in Road Scenes

Matias Valdenegro-Toro * **Paul Plöger** * **Stefan Eickeler** ** **Iuliu Konya** **

* *Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 Sankt Augustin, Germany (e-mail: matias.valdenegro@gmail.com, paul.ploeger@h-brs.de).*

** *Fraunhofer IAIS, Schloss Birlinghoven, 53757 Sankt Augustin (e-mail: stefan.eickeler, iuliu.konya@iais.fraunhofer.de)*

Abstract: Robust text detection and recognition in arbitrarily distributed, unrestricted images is a difficult problem, e.g. when interpreting traffic panels outdoors during autonomous driving. Most previous work in text detection considers only a single script, usually Latin, and it is not able to detect text with multiple scripts. Our contribution combines an established technique -Maximum Stable Extremal Regions- with a histogram of stroke width (HSW) feature and a Support Vector Machine classifier. We combined characters into groups by raycasting and merged aligned groups into lines of text that can also be verified by using the HSW. We evaluated our detection pipeline on our own dataset of road scenes from Autobahn (German Highways), and show how the character classifier stage can be trained with one script and be successfully tested on a different one. While precision and recall match to state of the art solution. A unique characteristic of the HSW feature is that it can learn and detect multiple scripts, which we believe can yield script independence.

Keywords: Text detection, Text recognition, Object detection, Object recognition, Histograms, Advanced Driver Assistance Systems.

1. INTRODUCTION

Textual information is relevant to robots and autonomous vehicles. In the context of autonomous driving, text gives information about the current geographical location, neighboring cities, towns and their distances, and relevant information about traffic, such as maximum speeds, nearby highway connections and exits. While precomputed and stored maps can provide much of this kind of information, they could be out of date and the vehicle has no way of validating such stored information. Temporary traffic panels can be installed to signal road works and deviations, which constitute local information that are not part of maps stored in a vehicle. This kind of information is valuable for autonomous driving, and in some cases necessary to avoid accidents. Autonomous vehicles and Advanced Driver Assistance Systems (ADAS) would greatly benefit from access to such information. This task requires text detection and recognition from visual sources in unconstrained environments.

Text detection (also known as text localization) is the problem of finding candidate text regions in an image for posterior classification. It should not be confused with text recognition, text detection is essentially a binary classification problem (determine if a region is text or not), while text recognition is a large multi-class (one for each character class) classification problem. Text detection is considerably harder than recognition due to large amounts of false positives that decrease precision and recall.

Many text detection algorithms exist (Zhang et al. (2013)) (Jung et al. (2004)). There are fundamental difficulties with text from natural scenes that are not present in other domains such as document processing. For example, lamp posts, windows, and bridges can easily confuse detection algorithms since they have shapes that are similar to characters.

Most text detection algorithms are developed and trained mostly for a specific language, which assumes a specific script (mostly Latin) from where characters are drawn. Such methods will fail when trained or tested with data from multiple scripts from other parts of the world (Kumar et al. (2013)), since they mostly use features that are only discriminative for Latin characters.

Vehicles can cross borders and traffic panels that use multiple scripts exist, like in Asian countries, as shown in Fig 1. This panel contains Latin and Chinese characters. In countries like India where many different languages with varying scripts are used, text detection and recognition is a much harder problem.

2. RELATED WORK

There is a rich literature about text detection (Zhang et al. (2013)), but only a small part of it is oriented

10.1016/j.ifacol.2016.07.716

Fig. 1. Multi-script Traffic Panel in Hong Kong

to multi-script detection. Detecting text is a difficult problem due to the high variability of text, for example, fonts, point sizes, word length, orientation, color, texture, script, and appearance. While the detection problem in printed documents is easier and already solved, detection in unconstrained images from natural scenes is harder and not yet solved.

Most of the existing methods for text detection can be classified into four categories:

Edge-based Filters are designed in order to extract edges that can be grouped to form characters and text lines (Neumann and Matas (2013)) (Liu and Samarabandu (2006)). Character grouping is usually done through heuristic rules that are hand-tuned for each application.

Texture-based Texture properties are used to discriminate text from non-text regions, usually through the use of a sliding window. Features are extracted for each region and a classifier is trained to discriminate and detect text (Minetto et al. (2013)). Performance is an issue with this kind of methods.

Region-based A region detection algorithm is used to extract regions from the image, and heuristic rules or a classifier is used to discriminate between text and non-text regions (Neumann and Matas (2012)) (Chen et al. (2011)).

Stroke-based Stroke width information is used to group pixels into characters, and heuristic rules are applied to discriminate and group text regions (Epshtein et al. (2010)) (Yao et al. (2012)).

In general, the use of heuristic rules is not appropriate for multi-script text detection, since the design of the rules incorporates assumptions or knowledge of a script. The use of such rules should be minimized to increase multi-script generalization.

Kasar et al. (Kasar and Ramakrishnan (2012)) use geometric, boundary, stroke, and gradient features with a Support Vector Machine (SVM) and neural network classifier to identify and group text components. This method obtains good results on a dataset of Latin and Indian scripts with curved text, but some of the features such as the convex deficiency are difficult and inefficient to compute. The large amount of engineered features that are used makes it difficult to adapt it into other scripts.

Gómez et al. (Gómez and Karatzas (2013)) use perceptual organization methods to cluster Maximally Stable Extremal Regions (MSER) that are first filtered using simple heuristic rules. Group hypotheses are generated and evaluated with an evidence accumulation framework. This method is multi-script and was evaluated on Latin and Korean. However the use of heuristic rules limits its generalization to other scripts.

Yin et al. (Yin et al. (2014)) use MSER regions, but instead of filtering the detected regions, they propose a filtering algorithm that works directly on the MSER tree, where regions with smaller variations in area are more likely to be characters. Distance metric learning is then used to learn and cluster character regions into text lines. This method performs well on a English and Chinese dataset, but the area variation assumption might not apply to some scripts.

In general, features are designed for particular scripts. There is no consensus about which feature is better for each script or if there are features that can be used to detect text independently of the script. Many text detection methods are complex to implement (Neumann and Matas (2013)) (Yin et al. (2014)) and have a high number of parameters. Reducing the number of parameters is generally beneficial as it is easier to train and less likely to overfit with a specific script.

3. TEXT DETECTION PIPELINE

Our proposed text detection algorithm involves four stages:

(1) **Region Extraction** The input image is converted to grayscale and MSER regions are extracted (Matas et al. (2004)).
(2) **Character Classification** Each detected MSER is classified as character or non-character region, and the latter regions are discarded. This stage is performed by Histogram of Stroke Widths (HSW) and a linear SVM classifier.
(3) **Text Line Grouping** Character regions are grouped to form horizontal text lines. This is implemented by raycasting between regions.
(4) **Text Verification** Text lines can either be output or post-processed with a text verifier. This stage uses the same HSW with a linear SVM classifier to discriminate real text regions from false positive ones.

We selected the MSER detector due to its robustness (Chen et al. (2011)). Since most characters in images have an almost constant color, the MSER detector usually detects these regions as connected components. In most traffic panels, there is high contrast between foreground and background which is very appropriate for a MSER detector. Other elements in the image might also be detected as regions, and a filtering step is required to remove non-character regions. Then, we aggregate neighboring characters to form text lines. We make the assumption [1] that text is nearly horizontal, which holds true for traffic panel text.

[1] Other text alignment directions can easily be added by either estimating the character orientation, or just including them in the algorithm.