Detecting Text in Natural Scenes with Stroke Width Transform

Boris Epshtein Eyal Ofek Yonatan Wexler

Microsoft Corporation

Abstract

We present a novel image operator that seeks to find the value of stroke width for each image pixel, and demonstrate its use on the task of text detection in natural images. The suggested operator is local and data dependent, which makes it fast and robust enough to eliminate the need for multi-scale computation or scanning windows. Extensive testing shows that the suggested scheme outperforms the latest published algorithms. Its simplicity allows the algorithm to detect texts in many fonts and languages.

1. Introduction

Detecting text in natural images, as opposed to scans of printed pages, faxes and business cards, is an important step for a number of Computer Vision applications, such as computerized aid for visually impaired, automatic geocoding of businesses, and robotic navigation in urban environments. Retrieving texts in both indoor and outdoor environments provides contextual clues for a wide variety of vision tasks. Moreover, it has been shown that the performance of image retrieval algorithms depends critically on the performance of their text detection modules. For example, two book covers of similar design but with different text, prove to be virtually indistinguishable without detecting and OCRing the text. The problem of text detection was considered in a number of recent studies [1, 2, 3, 4, 5, 6, 7]. Two competitions (Text Location Competition at ICDAR 2003 [8] and ICDAR 2005 [9]) have been held in order to assess the state of the art. The qualitative results of the competitions demonstrate that there is still room for improvement (the winner of ICDAR 2005 text location competition shows recall=67% and precision=62%). This work deviates from the previous ones by defining a suitable image operator whose output enables fast and dependable detection of text. We call this operator the Stroke Width Transform (SWT), since it transforms the image data from containing color values per pixel to containing the most likely stroke width. The resulting system is able to detect text regardless of its scale, direction, font and language.

When applied to images of natural scenes, the success rates of OCR drop drastically, as shown in Figure 11. There are several reasons for this. First, the majority of OCR engines are designed for scanned text and so depend on segmentation which correctly separates text from background pixels. While this is usually simple for scanned text, it is much harder in natural images. Second. natural images exhibit a wide range of imaging conditions, such as color noise, blur, occlusions, etc. Finally, while the page layout for traditional OCR is simple and structured, in natural images it is much harder, because there is far less text, and there exists less overall structure with high variability both in geometry and appearance.

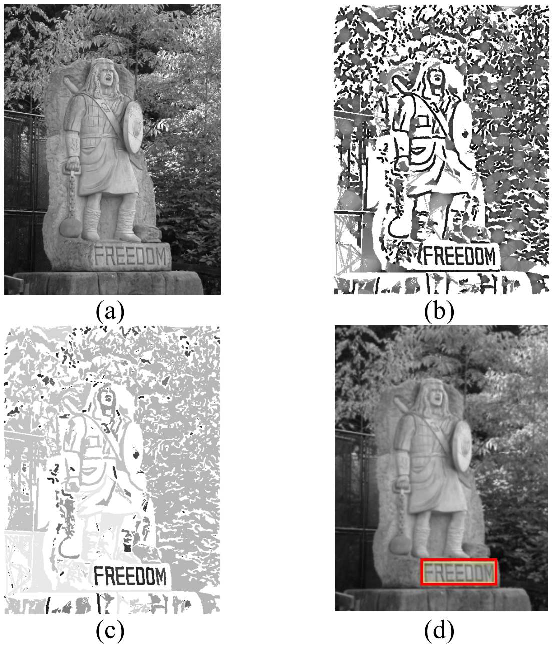

Figure 1**:** The SWT converts the image (a) from containing gray values to an array containing likely stroke widths for each pixel (b). This information suffices for extracting the text by measuring the width variance in each component as shown in (c) because text tends to maintain fixed stroke width. This puts it apart from other image elements such as foliage. The detected text is shown in (d).

One feature that separates text from other elements of a scene is its nearly constant stroke width. This can be utilized to recover regions that are likely to contain text. In this work, we leverage this fact. We show that a local image operator combined with geometric reasoning can be used to recover text reliably. The main idea presented in this work shows how to compute the stroke width for each pixel. Figure 1c shows that the operator output can be utilized to separate text from other high-frequency content of a scene. Using a logical and flexible geometric reasoning, places with similar stroke width can be grouped together into bigger components that are likely to be words. This reasoning also allows the algorithm to distinguish between text and arbitrary drawings as shown in Figure 2. Note that we do not require the stroke width to be constant throughout a letter, but allow slow bounded variations instead.

The method suggested here differs from previous approaches in that it does not look for a separating feature per pixel, like gradient or color. Instead, we collect enough information to enable smart grouping of pixels. In our approach, a pixel gradient is only important if it has a corresponding opposing gradient. This geometric verification greatly reduces the amount of detected pixels, as a stroke forces the co-occurrence of many similarly matched pairs in a small region. Another notable difference of our approach from previous work is the absence of scanning window over a multiscale pyramid, required by several other approaches [e.g. 3, 4, 25]. Instead, we perform a bottom-up integration of information, merging pixels of similar stroke width into connected components, which allows us to detect letters across a wide range of scales in the same image. Since we do not use a filter bank of a few discrete orientations, we detect strokes (and, consequently, text lines) of any direction.

Additionally, we do not use any language-specific filtering mechanisms, such as OCR filtering stage [3] or statistics of gradient directions in a candidate window pertaining to a certain alphabet. This allows us to come up with a truly multilingual text detection algorithm.

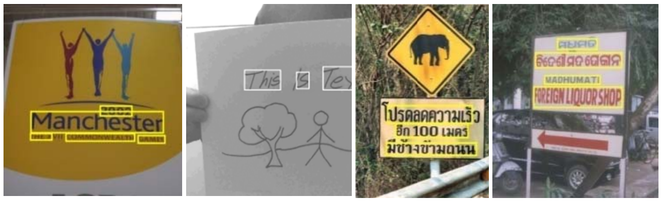

Figure 2: Detected text in natural images

Not every application of text detection requires a further step of character recognition. When such step is needed, a successful text segmentation step has great impact on the recognition performance. Several previous text detection algorithms [3, 18, 19] rely on classification of image regions and therefore are not providing a text segmentation mask required for subsequent OCR. Our method carries enough information for accurate text segmentation and so a good mask is readily available for detected text.

2. Previous work

A great number of works deals directly with detection of text from natural images and video frames. Related works from other domains study the extraction of linear features.

For comprehensive surveys of methods for text detection, see [1, 2]. In general, the methods for detecting text can be broadly categorized in two groups: texture-based methods and region-based methods. Texture-based methods [e.g. 3, 4, 18, 19, 22] scan the image at a number of scales, classifying neighborhoods of pixels based on a number of text properties, such as high density of edges, low gradients above and below text, high variance of intensity, distribution of wavelet or DCT coefficients, etc. The limitations of the methods in this category include big computational complexity due to the need of scanning the image at several scales, problems with integration of information from different scales and lack of precision due to the inherent fact that only small (or sufficiently scaled down) text exhibits the properties required by the algorithm. Additionally, these algorithms are typically unable to detect sufficiently slanted text.

Another group of text detection algorithms is based on regions [e.g. 5, 6, 23]. In these methods, pixels exhibiting certain properties, such as approximately constant color, are grouped together. The resulting connected components (CCs) are then filtered geometrically and using texture properties to exclude CCs that certainly cannot be letters. This approach is attractive because it can simultaneously detect texts at any scale and is not limited to horizontal texts. Our method falls into this category, but the main feature that we use is very different from the typically used color, edges or intensity similarity. We measure stroke width for each pixel and merge neighboring pixels with approximately similar stroke width into CCs, which form letter candidates.

The work that uses a somewhat similar idea of detecting character strokes is presented in [7]. The method, however, differs drastically from the algorithm developed in this paper. The algorithm proposed in [7] scans an image horizontally, looking for pairs of sudden changes of intensity (assuming dark text on bright background). Then the regions between changes of intensity are examined for color constancy and stroke width (a range of stroke widths is assumed to be known). Surviving regions are grouped within a vertical window of size *W* and if enough regions are found, a stroke is declared to be present. The limitations of this method include a number of parameters tuned to the scale of the text to be found (such as vertical window size *W*), inability to detect horizontal strokes, and the fact that detected strokes are not grouped into letter candidates, words and sentences. Consequently, the algorithm is only able to detect near-horizontal text. The performance results presented in the paper are done using a metric that is different from the ICDAR competition