Text Detection on ISPs

James Hegarty (jhegarty@stanford.edu)

May 2, 2012

Description and Goals

Text detection takes as input an image of a natural scene and returns binarized crops of any text present in the scene. This is a crucial first step for a number of augmented reality applications such as Word Lens.

ISPs (Image Signal Processors) process the raw data off the image sensor in digital cameras. Current ISPs only support fixed-function operations like demosaicing and denoising. However, in my research I'm developing a new architecture for programmable ISPs which is designed to support algorithms like text detection. Algorithms that run on an ISP are limited by the following constraints:

1. ISPs perform computations on sensor data prior to writing it to memory. So, the algorithm can only access a subset of the image that it is able to buffer on chip (typically ∼10 lines).

2. The algorithm must exploit spatial locality and data parallelism to attain high throughput.

In this project I will attempt to develop a text detection algorithm that fits within these constraints.

Planned Work

First, I will implement the MSER, geometric filtering, and stroke width filtering stages of the text detection algorithm described by Chen et al. [3]. Then, I will implement a new text detection algorithm that is similar to [3] but replaces individual components of the algorithm with algorithms that are more amenable to data-parallel execution. For example, I will replace MSER with Shafait's local adaptive thresholding [2]. It will be challenging to have the algorithm detect text that covers a large area in the image given the limited buffer size. However, I think I can work around this by building a gaussian pyramid on the fly and implementing text detection as an online algorithm that performs computations every time a new line is generated in the pyramid.

To evaluate my algorithm, I will compare its recall and precision to Chen et al. [3] using a standard dataset [1], and estimate whether its memory usage and op counts are reasonable for an algorithm running on an ISP.

Platform

I will not be using an Android camera phone - current ISPs in phones aren't sufficiently programmable and don't expose an API to the programmer. Instead, I will implement Chen et al. [3] on the CPU in Matlab, and implement my new algorithm in a C simulator that imposes the ISP constraints and allows me to capture relevant statistics.

Possible Extensions

I would like to implement the letter pairing, line formation, and word separation algorithms from Chen et al. [3] on the CPU and pipe the results to a commercial OCR system. However, these features are not required to evaluate or prove the viability of the text detection algorithm itself, so I will only implement these features if I have extra time.

References

[1] Lucas, S.M. *ICDAR 2005 text locating competition results.* Eighth International Conference on Document Analysis and Recognition (2005), pp. 80-84 Vol. 1.

[2] Faisal Shafait, Daniel Keysers and Thomas M. Breuel. *Efficient implementation of local adaptive thresholding techniques using integral images.* Proc. SPIE 6815, 681510 (2008).

[3] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod. *Robust text detection in natural images with edge-enhanced maximally stable extremal regions.* IEEE International Conference on Image Processing (ICIP), September 2011.