Project Proposal for the Perceptual Computing Major Project

# Title and Author

Automatically De-wrap, Transform, Crop and Colour Reduce Images of Documents.

By Christopher Skorka

# Introduction

The goal is to write a python program that accepts images of documents (i.e. images taken with a camera), detects the main document in the image, its corners and edges to then de-wrap and transform the image to flatten the document and crop it using corner and edge data. Lastly It should convert the resulting image into a black and white image using Otsu’s method.

Previously existing page de-wrapping algorithms work by detecting lines of text and calculating the transformations required to straighten these lines. However, this requires horizontal text and is less suitable for documents containing a significant portion of vertically oriented text and graphics. Thus, this project explores a combination of edge and corner detection to solve this problem. The program aims to identify the page as a whole and successfully de-wrap the image regardless of content.

# Technical Approach

## Gaussian smoothing

Likely gaussian smoothing will be required before applying any higher-level operations such as edge detection, corner detection and colour thresholding.

## Corner Detection

The corners of the page need to be detected. The program will need to find all courses and determine the quadruplet that form the outline of the page

## Edge detection

The canny edge detection techniques will be required to find one-pixel wide edges and their orientation. A continuous loop of edges that pass through the 4 corners of the page is selected and used to calculate the transformation operations required.

## Image transformations

De-skew and transform the image such that the edges of the page are straight and the corners of the page are in the corners of the image.

## Colour thresholding

Otsu’s method will be used to turn the image black and white. Since there may be shadows or shadow gradients over the image, Otsu’s method would be applied in windows across the image and the resulting data would be combined to produce a uniform back and white document image.

# Milestones

1. Basic IO – loading images into workable formats, displaying them with additional information such as corners and edges detected and storing them.
2. Corner detection – Detect corners, their orientation and size.
3. Filter out the corner quadruplet that roughly represents the outline of the page (excluding skewing and bends).
4. Detect edges – Detect edges, their direction and intensity.
5. Filter out edges that pass through the four corners and represent the outline of the page
6. Calculating transformation matrices and operations necessary.
7. Perform Transformation to de-skew, crop image.
8. Apply Otsu’s method to turn the document black and white using an appropriately selected threshold.

# References

|  |  |
| --- | --- |
| [1] | Baruch, Robert. “Dewarping Pages.” *The Half-Baked Maker*, 27 Feb. 2010, [www.halfbakedmaker.org/blog/366](http://www.halfbakedmaker.org/blog/366). |
| [2] | Zucker, Matt. “Page Dewarping.” *Needlessly Complex*, 15 Aug. 2016, <mzucker.github.io/2016/08/15/page-dewarping.html>. |
| [3] | Kim, Chelhwon, et al. “Dewarping Book Page Spreads Captured with a Mobile Phone Camera.” *Camera-Based Document Analysis and Recognition Lecture Notes in Computer Science*, 2014, pp. 101–112., doi:10.1007/978-3-319-05167-3\_8. |