Robust Automatic Detection of a Document in an Image

Christopher Skorka
Griffith University
christopher.skorka@griffithuni.edu.au

Abstract

This paper presents an approach for robust detection of documents in images taken with cameras. The proposed detection system combines variable image prepossessing, Harris corner detection, canny edge detection, Hough line transformation and a steepest ascent search algorithm to detect the page with increasing confidence in an iterative process. This approach is moderately successful with example images of documents that have a relatively high contrast to their background and are the main subject of the image. This approach outperforms other contour-based approaches.

1. Introduction

Since the dramatic increase in high quality camera found on affordable modern smart-phones it has become appealing to photograph documents rather than using a document scanner. Thus, the need for document recognition and extractions has arisen to allow for users to extract documents from images with scanner like quality. Previous research investigates the use of edge detection, contour findings with carefully selected parameters for said processes to produce the desired results. The proposed approach combines a number of image preprocessor techniques, Harris corner detection, Canny edge detection, Hough line transformation, line grouping and a steepest ascent local search algorithm to detect the document with increasing confidence in an iterative process. This approach finds corners and straight edges, calculates edge intersections, compares them to the corners. From this data the most likely page outline is computed and selected. Next confidence measure of the overall process and outcome is calculated and the local search algorithm tweaks the parameters for the above process iteratively until a local maximum is found and the most confident finding is returned.

1.1. Previous work

Many implementations of document detection in images use Canny edge detection followed by finding contours from the edge detection response. A number of online blogs and tutorials from Rosebrock [2], Sharma [3], Jackson [4] and Hajek [5] use this approach, however, due to the nature of canny edge detection and contour detection it is prone to failure when glare blurs edges, other edge outside the document are present or a corner o the document is slightly outside of the image. This is demonstrated in 6 Experiments and results where it is shown that Hough line based approaches are better suited to solve this task.

1.2. Overview

To achieve accurate results data from Harris corner detection, Canny edge detection and Hough transform are combined to find possible bounding boxes of the document and compute a confidence measure. To compute possible bounding boxes the Hough line transformation is applied to the Canny edge response, lines are groups and, possible rectangles from quartet of liens are generated and rectangle vertices are compared to Harris corner detection response to compute a confidence value for each possible bounding rectangle. Through trial and error good parameters were found for these processes that produce relatively reliable results for the generated test cases. The found parameters for said operations are used as a starting point and are adjusted and optimized by a steepest ascending local search algorithm. This state space search algorithm takes an overall confidence measure as its input. This overall confidence value is based on the combined confidence values of each possible rectangle found using the tested parameters. Using this system we are able to detect documents in images more reliably that the previously existing contour based approach.

2. Preprocessing

Before applying high level operations on the image, it is preprocessed to increase the accuracy of future operations. To begin with the image is smoothed using a gaussian filter to reduce noise. Next a closing operation is performed to reduce corner and edge detection rate on text and graphics. The combinations of these processes prepare the image for the following higher-level operations and final the document detection.

2.1. Smoothing

A standard gaussian low pass filter is applied over the image to reduce noise. Noise produces increased differential values in the image that incorrectly trigger edge and corner detection. The removal of this noise is essential for the higher-level operations as it decreases incorrect triggering of edge and corner detection that may obscure the results. However, it is important not to exaggerate the smoothing effect to preserve real edges and corners that are essential in following operations. Thus, a small kernel that's of size 3 by 3 with the gaussian distribution with standard deviation is used to achieve the desired results. Increasing the kernel size to 5 by 5 and beyond combined with a standard deviation to 2 or beyond dramatically decreased the systems overall accuracy and ability to detect documents.

2.2. Closing

Closing is the process of dilating and then eroding the image to close off holes or dark spots in the image. It is applied to reduce text from triggering edge and corner detection without altering the edges and corners of the document itself. The dilation and erosion are performed with identical structuring elements. The kernel a circular one with a default size of 5 by 5. This size has been very reliable with an increase usually resulting in little to no improvement overall. The circular kernel is selected as it produces fewer edges and corners than a rectangular kernel does. This closing procedure reliable prevents edge and corner detection of the text on the document and preserves the outline of the document for following operations as can be seen in figure 1.

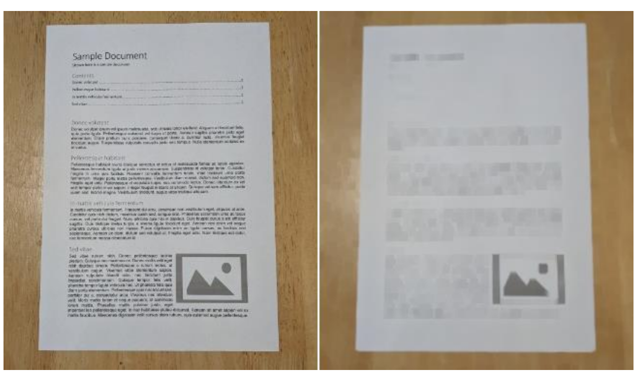

Figure 1: closing results with a 5 by 5 kernel (right) from source (left).

3. Higher level operations to gather edges and corners

Once the image preprocessing is complete higher-level operations are done to gather data and find the documents outline. These operations include Harris corner detection, Canny edge detection and Hough transformations. The Hough transformations are applied to the Canny edge output to find major edges. Pairs of horizontal and vertical lines are used to calculate the 4 intersection points which are compares to the Harris corner detection output. This data is then combined to compute a confidence measure. This process often returns multiple possible document outlines with varying confidence levels of which the one with the greatest confidence is selected.

3.1. Harris corner detection

Harris corner detection is performed on the resulting image after the preprocessing steps. The corner data found in this step is used later to compute the confidence of possible document outlines found by matching document corners to corners found in this step. The default block size of 25 to detect larger corners and a default threshold of 0.03. This threshold usually varies significantly between different test images. Of course, the corner detection responds to corners other than those of the page, thus, the threshold needs to be set such that the page corners are detected whilst detecting as few other corners as possible.

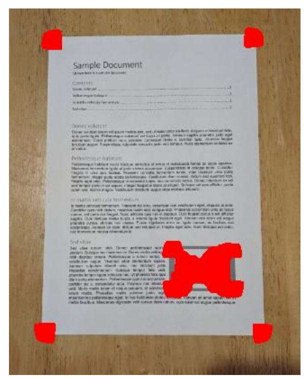

Figure 2: Harris corner detection response.

3.2. Canny edge detection

Canny edge detection is also performed on the image preprocessing result. If the thresholds are selected appropriately it should be able to detect at minimum the outlies of the document but usually graphics, some text and shadows produce additional edge responses. The results in figure 3 show distinct outline of the document with a gap and edges from a graphics being detected as well. Small gaps are usually not enough to disrupt the Hough line transformation and are acceptable. Some graphics and text generated edges are generally not an issue so long as they are relatively small in size. A problem arises however, when the image has a very low contrast. Then text or shadows may be detected at a significantly lower threshold than the document outlines. If this occurs the system will struggle to detect the page.

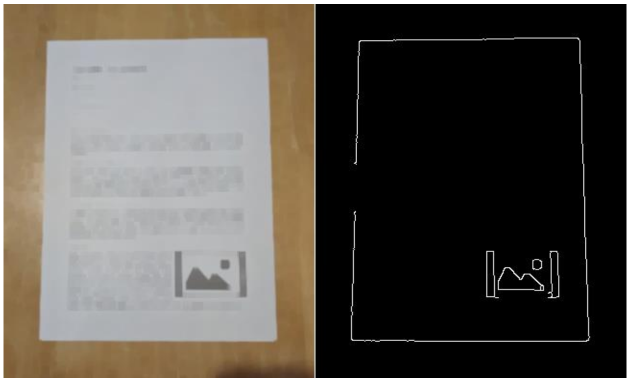

Figure 3: Canny edge detection response (right) from preprocessed image (left).

3.3. Hough transform

The Hough transform is applied to the canny edge output to find major lines in the image. Line quadruplets will then be used to find possible quartets of corners that may represent the document outline. The default threshold for the Hough transform is 60 with the rho and theta thresholds being 1. These values produce the results seen in figure 4. In this example the edges of the document are detected but multiple responses are generated for each edge of the document. Often there exists no threshold for the Hough transform and Canny edge detection that allows for the detection of exactly one line per edge and for document edges only. In figure 5 the threshold was reduced to include all edges of the document which caused many responses for the document edges as well as other edges in the image. To solve this problem similar lines are combined into one in a process described later. Note in figure 4 with the default thresholds the graphics in the document is not detected by the transformation but in figure 5 with lowered thresholds it is detected.

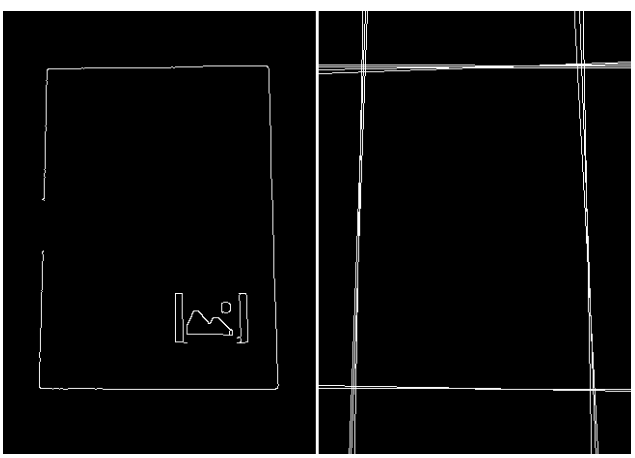

Figure 4: Hough transformation result (right) from Canny edge detection (left).

3.4. Filtering Hough line results

As mentioned previously the Hough line transform responds multiple times for the same edge which unwanted in addition to possibly responding edges that are not the document edge. To resolve this issue two steps are taken. First horizontal and vertical lines are filtered. This is done by applying some threshold to the angle of the line. Secondly similar edges are combined into one. The default angle threshold is roughly 30 degrees to allow for perspective distortions. Secondly lines are grouped together based on a distance and angle threshold. Out of a group of lines the center line is selected. The default values for the grouping is 10 and ~5 degrees for the rho and theta thresholds respectively. Figure 5 demonstrates the results of horizontal and vertical line filtering combined with grouping. As can be seen the number of possible rectangular outlines are reduced significantly. Note that the Hough transformation threshold needed to be significantly lower in figure 5 thus producing the large number of responses as seen.

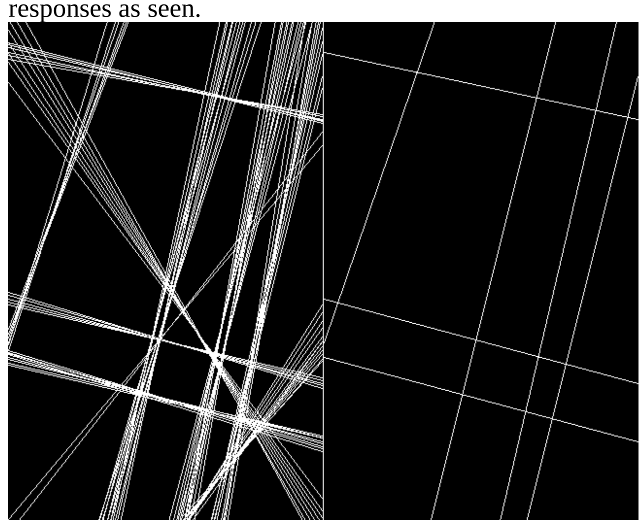

Figure 5: Line reduction (right) from Hough line transformation (left).

4. Combining edge and corner data

In order to detect the final outline of the document all possible bounding rectangles are found from the filtered Hough lines and a confidence measure is computed with the Harris corner detection results in consideration. The rectangle with the greatest confidence will be selected as the final document outline.

4.1. Finding possible outlining rectangles

Once the Hough lines have been filtered all possible rectangles are found from these liens. To find each rectangle each combination of vertical and horizontal line pairs is combined to produce 4 lines that from a roughly right-angled rectangle. The corners of each found rectangles are calculated and a list of possible rectangles is composed. Figure 6 shows these rectangles, note that multiple rectangles are found.

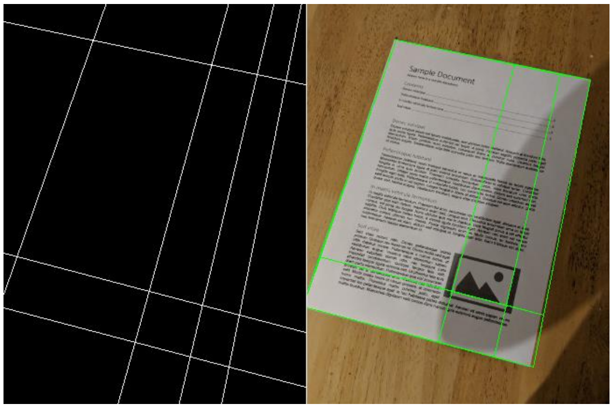

Figure 6: Possible bounding rectangles (right) from filtered Hough line transformation (left).

4.2. Computing bounding rectangle confidence

In order to select one rectangle of the many found some confidence value needs to be computed to compare the rectangles. Two good properties are area and number of corners that overlap with the results from the Harris corner detection. In general, the larger the rectangle to more likely it is to be the page. Additionally, it is expected that Harris corner detection is more likely to detect the document corners than smaller corners of content contained within. Thus, if the found rectangle's corners overlap with the Harris results the confidence shall increase. The confidence is computed by the following formula:

$$confidence = \frac{rectArea}{imgArea} \times \frac{matchedCorners+1}{5}$$

With this formula applied to each rectangle to most probable rectangle can be selected as shown in figure 7.

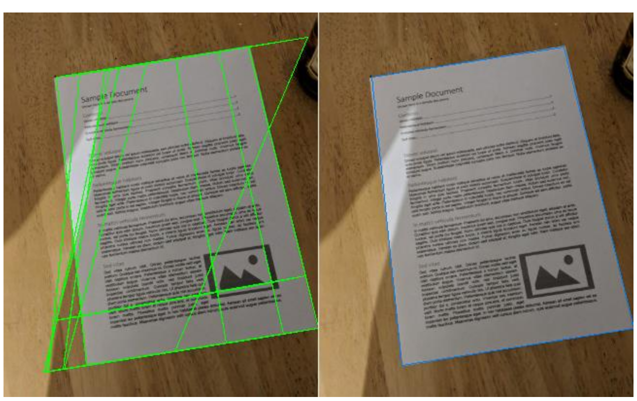

Figure 7: Most like rectangle selected (right) from all possible rectangles found (left).

5. Optimizing parameters

Since each photo is taken in a different environment the parameters for the preprocessing, Harris corner detection, Canny edge detection, Hough transform and line grouping need to be adjusted for the given circumstances. To optimize these values a steepest ascent local search algorithm refines the values and computes an overall confidence value for the parameters. The overall confidence is computed by dividing the average confidence value of all found rectangles by the number of rectangles. The formula is as follows:

$$overallConfidence = \frac{\sum rectangleArea}{rectangleCount^2}$$

This allows us to iteratively refine the parameters of for the processes mentioned previously and produce increasingly confident results.

6. Experiments and results

To test the robustness of this system, it has been tested in a number of different test images taken in different lighting conditions, angles and with different backgrounds. Shown bellow are some of the successful results in figures 8-12 (the images are ordered as follows: original, closed, Harris corner detection, Canny edge detection, Hough line transform, filtered lines, possible rectangles and selected document outline):

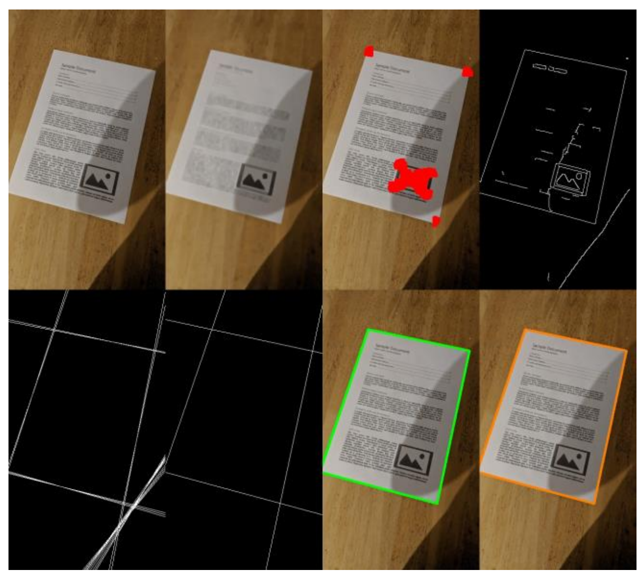

Figure 8: Successful detection with one corner outside.

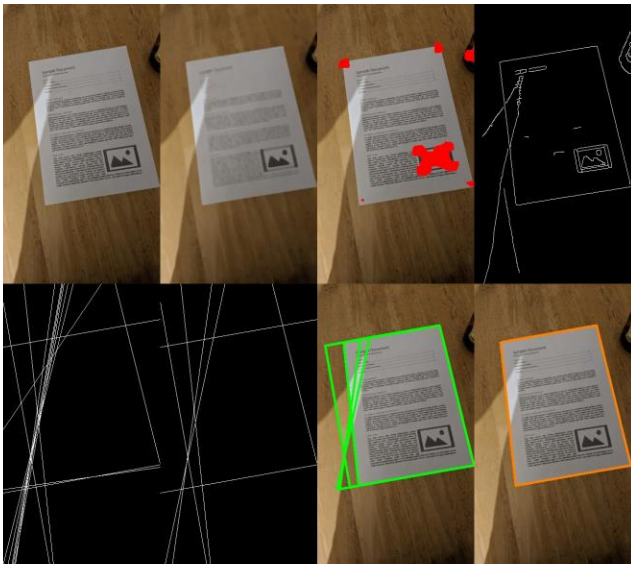

Figure 9: Successful detection with obscuring shadows.

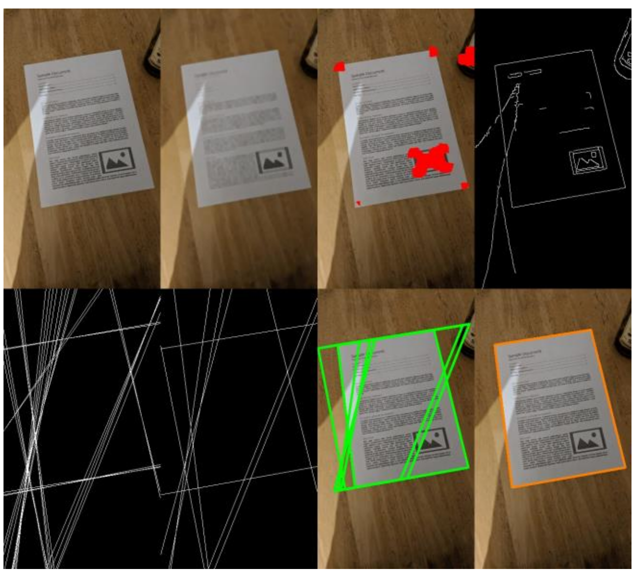

Figure 10: Successful detection with obscuring shadows.

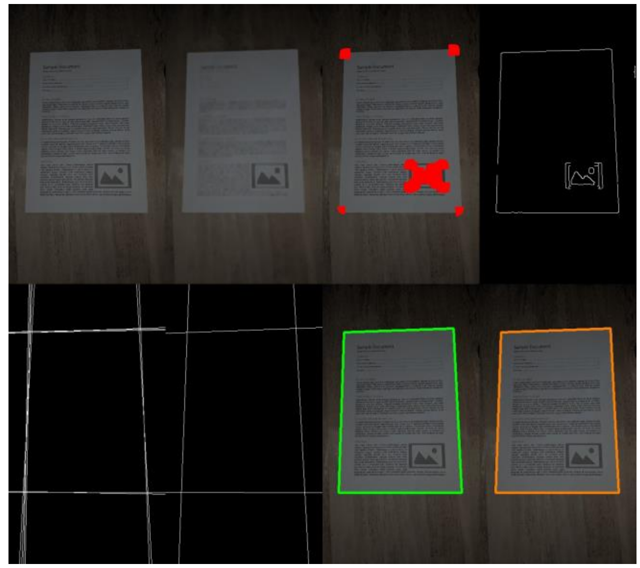

Figure 11: Successful detection in low light.

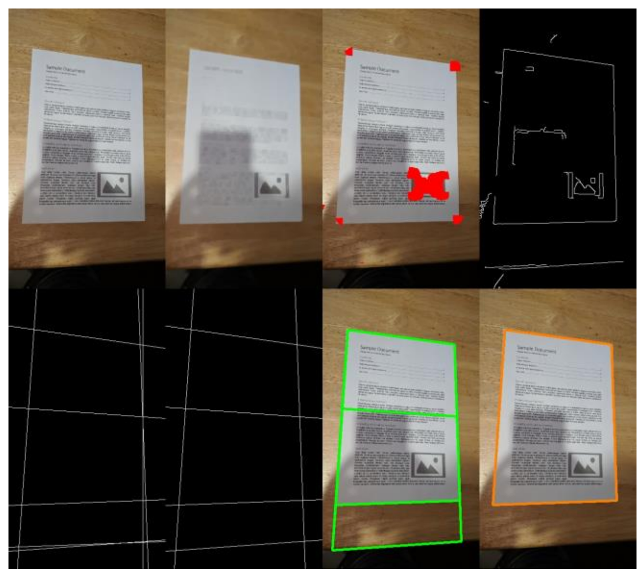

Figure 12: Successful detection with misleading edges.

As can be observed in figures 8-12 this approach is able to detect the document in various conditions ranging from different lighting conditions, document orientation, camera position. However, the approach fails sometimes with some distinct edges present or if there is no sufficient contrast between the document and the background as can be observed in figures 13-14:

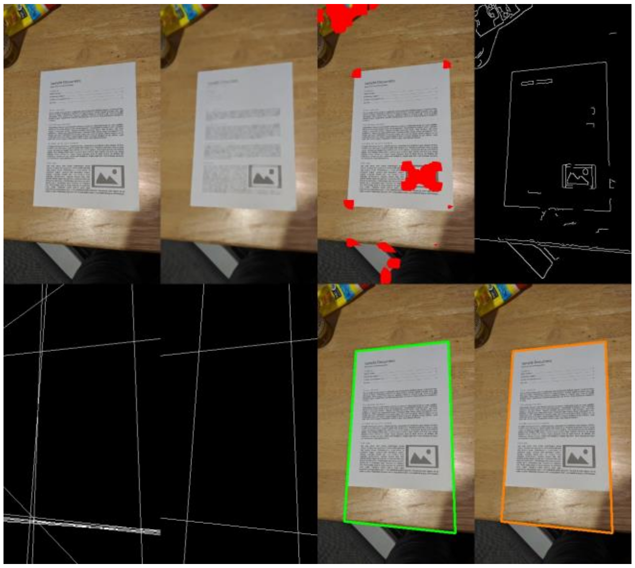

Figure 13: Edge of the table combined with glare produce inaccurate results.

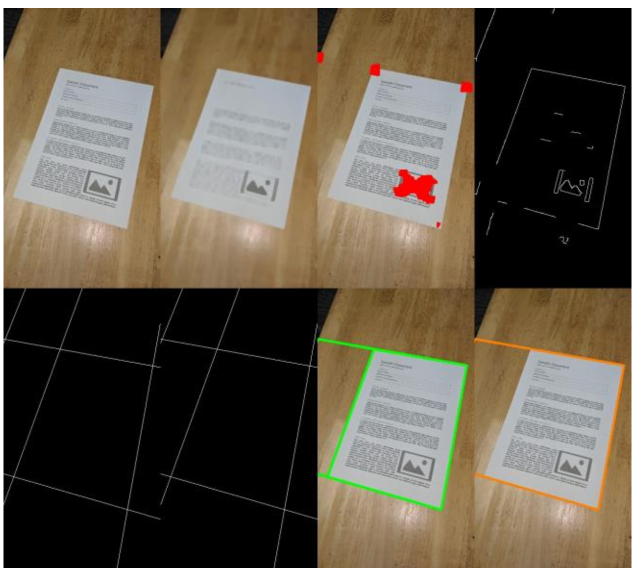

Figure 14: Edge of the table causes incorrect detection.

Figure 15: Low contrast causes no Harris corner response and misleading edge responses.

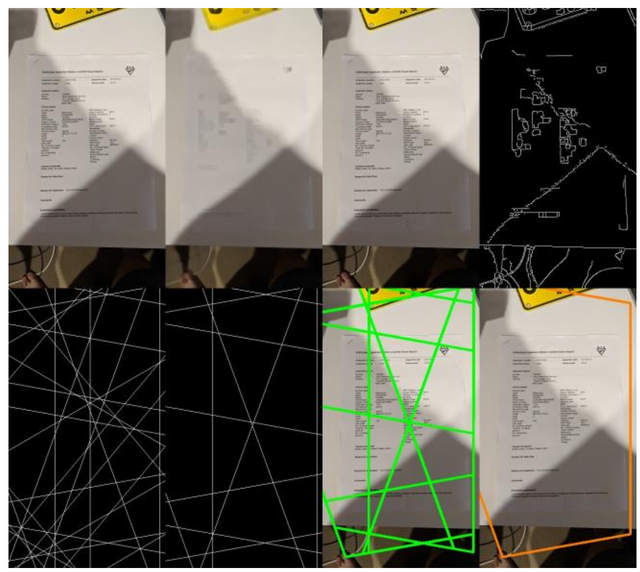

Figure 16: Low contrast causes no Harris corner response and misleading edge responses.

As seen in figures 13 and 14 the table edge together with glare obscures the data used to detect the document outline and thus producing inaccurate results. This misclassification is due to insufficiently accurate confidence measure of the predictions made. By improving this measure to include other factors such as the response rate of the higher-level operations and analyzing the border produced the predictions may increase accuracy. In figures 15 and 16 the program is presented with a very difficult situation, the document has low contrast to the background, string shadows cover the image and distracting objects are visible at the top and bottom of the image. This is a result of the Harris corner detection not detecting the document corners at all and Canny edge detection finding many edges before detecting the documents outline. Thus, images like figure 15 and 16 with low contrast between the document and background and significant number of distractions are not suitable for this approach.

To compare this approach to the previously existing approaches involving contour detection this system's Hough line output was compared to the contour detection in figures 17-19 (Image order is: Canny edge response, Hough line transformation and contours with 4 vertices).

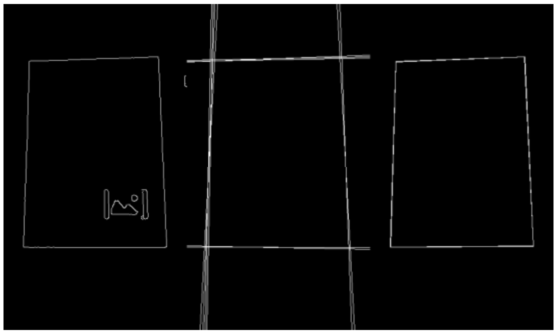

Figure 17: Clear Canny response results in good contour.

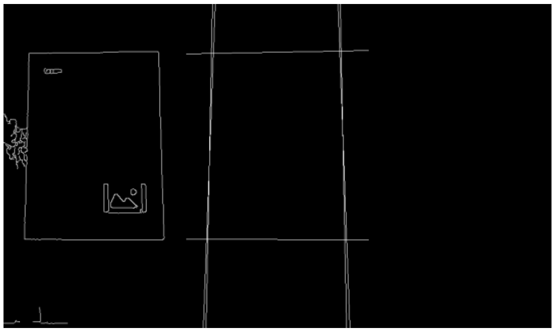

Figure 18: Less clear Canny response with the document outline still visible results in no contours with 4 vertices.

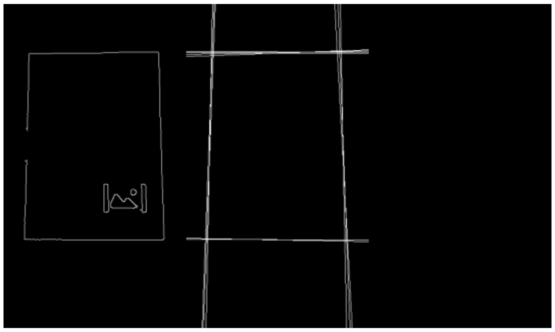

Figure 19: Broken document outline results in no contour whilst Hough lines respond appropriately.

As can be seen in figures 17-19 contours are not reliably detected whilst Hough lines are better able to detect the document outline despite fuzzy Canny edge responses or broken edges due to glare or other reasons. Thus, the Hough line oriented approach has a greater likely hood operating successfully.

In order to test whether the high-level operations parameters can be tweaked manually to improve results, a real time parameter adjustment interface has been implemented to manually test parameter combinations and search for possible improvements to detection accuracy.

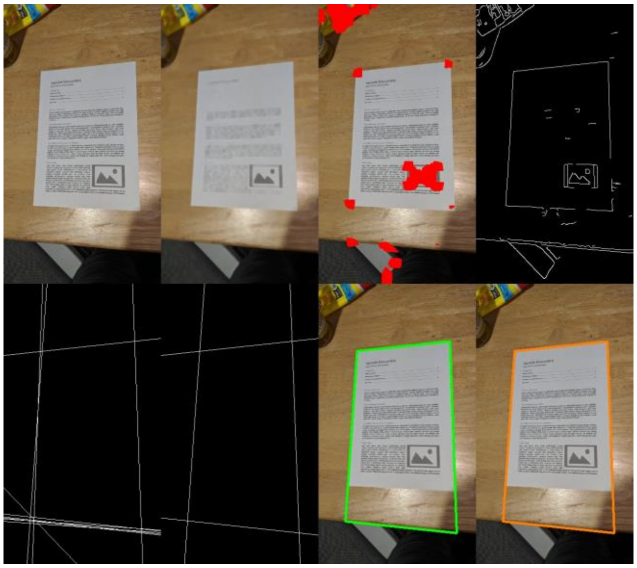

Figure 20: Example automated output.

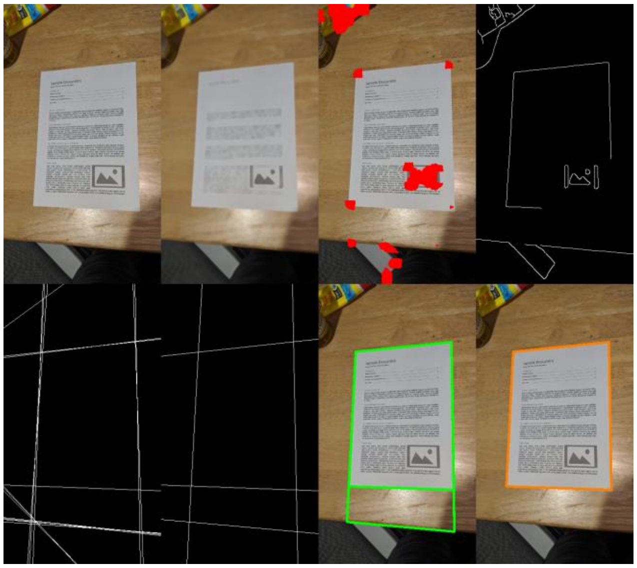

Figure 21: Manually adjusted parameters allows for successful the detection of the document.

Figures 20 and 21 show that this approach fails when previously mentioned unwanted features are present, however the Canny edge detection, Harris Corner detection and Hough line transformatin parameters can be tweaked such that the document is identified successfully. This indicates improvements to the confidence measure calculations are required to allow the state space seach algorithm and voting system to identify the correct boundries.

Finding workable parameters for figures 15 and 16 was unsuccessful. This suggest that the fundamental operations for this approach are not suitable for such cases and improvements to the confidence measures and parameter state space search algorithm will likely not improve accuracies in this domain.

7. Conclusion

This approach for detecting documents in images is partially successful. It is able to reliably detect the document in images in various lighting conditions and orientations so long as there are little to no major distracting features such as table edges and other objects and contrast between the document and the background is significant enough.

This system is open for future improvements. In particular the confidence measure for measuring possible document boundaries significance needs to be improved to better reflect the probability of a detection being the true document.

References

[1] 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.

[2] Rosebrock, Adrian. "How to Build a Kick-Ass Mobile Document Scanner in Just 5 Minutes." *PyImageSearch*, 1 Sept. 2014, www.pyimagesearch.com/2014/09/01/build-kick-ass-mobile-document-scanner-just-5-minutes/.

[3] Sharma, Vipul. "Document Scanner Using Python + OpenCV." *Vipul Sharma*, 9 Jan. 2016, vipulsharma20.blogspot.com/2016/01/document-scanner-using-python-opencv.html.

[4] Jackson, Brian. "Scanning and OCR-Ing a Paper Receipt." *Brian Jackson*, 2 Jan. 2016, www.astrojack.com/scanning-and-ocr-ing-a-paper-receipt/.

[5] Hajek, Breta. "Scanning Documents from Photos Using OpenCV." *Breta Hajek*, 25 Jan. 2017, bretahajek.com/2017/01/scanning-documents-photos-opencv/.

[6] Stamatopoulos, N., et al. "Page Frame Detection for Double Page Document Images." *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems - DAS '10*, 2010, doi:10.1145/1815330.1815382.

[7] Canny, John. "A Computational Approach to Edge Detection." *Readings in Computer Vision*, 1987, pp. 184–203., doi:10.1016/b978-0-08-051581-6.50024-6.

[8] Zucker, Matt. "Page Dewarping." *Needlessly Complex*, 15 Aug. 2016, mzucker.github.io/2016/08/15/page-dewarping.html.

[9] Baruch, Robert. "Dewarping Pages." *The Half-Baked Maker*, 27 Feb. 2010, www.halfbakedmaker.org/blog/366.

[10] Petkovic, Tomislav, and Sven Loncaric. "An Extension to Hough Transform Based on Gradient Orientation." *Proceedings of the Croatian Computer Vision Workshop, Year 3*, 22 Sept. 2015.

[11] Ballard, D.h. "Generalizing the Hough Transform to Detect Arbitrary Shapes." *Pattern Recognition*, vol. 13, no. 2, 1981, pp. 111–122., doi:10.1016/0031-3203(81)90009-1.

[12] Olson, Clark F. "Improving the Generalized Hough Transform through Imperfect Grouping." *Image and Vision Computing*, vol. 16, no. 9-10, 1998, pp. 627–634., doi:10.1016/s0262-8856(98)00083-3.