Using a Generic Camera as a Scanner

Colin Schmidt, Nicholas Sunjaya, Yi Hong To
UC Berkeley
{colins, nicholas.sunjaya, craigyihongto}@berkeley.edu

Abstract

In this paper we describe a method to turn ordinary pictures of books and papers into scanner-like images. This transforms and aligns the image such that the page takes up the entire viewing area. This makes the image more readable and enables further processing such as character recognition. We present the results of running our algorithm on a hand-annotated set of 50 images. We discuss the successes failures and future work for our algorithm.

1. Introduction

Scanning documents has been a common practice for a long time. Even before the advent of digital scanners faxes were a popular of sending documents to others. But as smartphones have become more ubiquitous and fully featured they have begun to absorb many features of other electronics. XXXXX TODO:CITE TECHONOLGY STUDY XXXXX

In this paper we present an algorithm for turning a simple smartphone camera, or any other camera, into a scanner. The properties of a scanner we will replicate are: the page scanned fills the image, it is orientated with the top of the page at the top of the image, and that the ratio of the image is the same as that of the paper.

We first discuss previous work in this area in section 2. We then describe our motivation for developing this tool in section 3. Next, we introduce our multi-step algorithm with illustrations in section 4. Following this is a description of our experimental methodology and the results of applying the algorithm to a set of images in section 5. Then, we discuss which images are problematic for our algorithm and potential solutions in section 6. Finally, in section 7 we conclude with an exploration of future work and highlight out successes.

2. Related Work

This section describes other approaches and implementations. We should probably cite them like this [1].

3. Motivation

Sending documents to others digitally has huge advantages over physical mail. It is nearly instaneous, the recipient can easily share it with others, and you can often search through the document for particular words and phrases. However, some things are still printed and sent physically including books, notes, and bills. Therefore there is a need for the ability to turn physical papers into digital files. For several decades this was accomplished with a special machine, the scanner, also sometimes integrated into high-end printers.

Recently cell-phone cameras have become much higher resolution. This has enabled a new opportunity, to allow these camera's people have with them at all times to be turned into scanners. The added convience will allow for the convience of digital transmission for all documents even without access to a potentially expensive and/or large scanner.

A cell-phone application can also offer an easy interface to the user. Printers and scanners can be notoriously fussy so moving to an easy to update and change UI/UX could greatly improve the usability.

4. Algorithm

Our algorithm is a multi-step process to identify the bounding box of the item and then a homography transform to produce the final image. We will now walk through the algorithm with a sample image seen in figure 1.

The first step of the algorithm is to identify edges in the image. In our case we first smooth the image with a gaussian filter to help remove some background noise. Then we use a canny edge detector to give us the binary image of edges. Because of the non-maximal supression the canny edge detector gives us the best chance of getting fine lines. Fine lines enable the next steps in our algorithm to be more precise. The output of the edge dection can be seen in figure 2.

Once we have the binary image of edges we need to detect which edges correspond to straight lines. To accomplish this we use the Hough transform [**?**]. TODO: explain

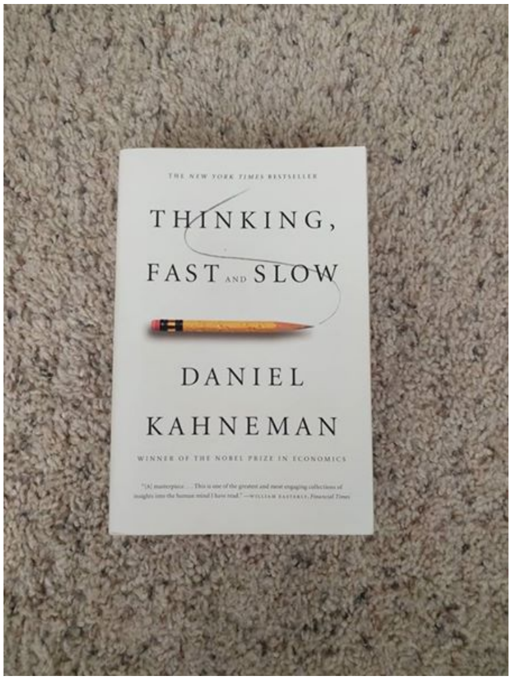

Figure 1. What the image looks like before any modification.

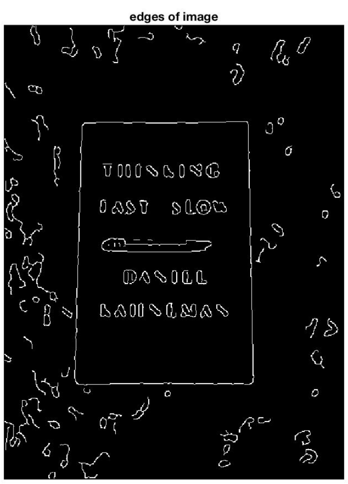

Figure 2. After edge-detection.

in more detail how Hough works. This transformation gives us up to 10 line candidates. These prospective lines can be seen in figure 3

Next we use a random sampling method inspired by RANSAC to pick the lines that best represent the item. In this step we use some assumptions about the images to develop a heuristic for which lines should be best. We assume that the largest quadrilateral in the image is the item, such

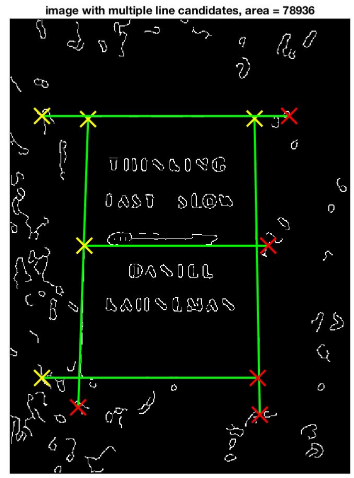

Figure 3. The second step

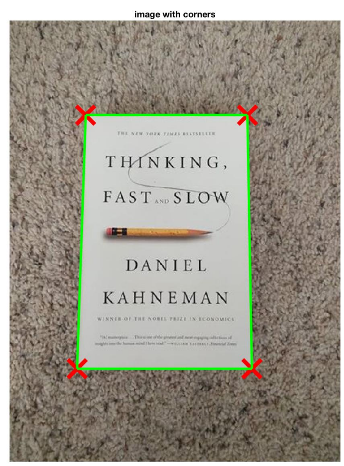

Figure 4. The third step

that maximizing the area of our polygon will be the correct bounding box for our item. We also assume that it will be approximately rectangular, such that lines forming near right angles are preferred over acute or obtuse angles. Using this criteria we then randomly sample four lines from the set keeping track of which best meets our heuristic. At the end of this process we have four lines we believe to lie on the bounding box of the item. With these lines we then find their intersection and mark these as the corners of the item. The result of this step can be seen in figure 4.

Finally we use this set of corners as the input to a homog-

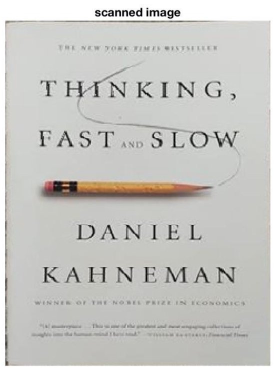

Figure 5. The fourth step

raphy transformation. Once we have computed the transform we then transform all pixels in the bounding box to get the final image as seen in figure 5.

5. Results

We tested our algorithm on a set of 50 manually annotated images of books, and papers. The image set had a variety of backgrounds, colors, and lighting. We tuned the parameters of our algorithm across all the images such that for the final test run we used the same parameters for all images.

We categorized images as successes or failures based on the detected corners compared to the annotated corners. Results that were within a small margin of the annotated corners counted as successes and otherwise were failures. This resulted in an accuracy of 71% in our image set. In the following section we will discuss the failure modes of our algorithm.

6. Discussion

We'll talk about some of the problems we had and potential future solutions.

7. Conclusion

Our algorithm works ok. It could do with some improvement but we leave that as future work.

References

[1] Authors. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as additional material `fg324.pdf`.