Extraction of Handwriting in Tabular Document Images

Robert T. Clawson
Brigham Young University
Department of Computer Science
Provo, Utah, USA
rtclawson@byu.edu

William A. Barrett
Brigham Young University
Department of Computer Science
Provo, Utah, USA
barrett@cs.byu.edu

ABSTRACT

We propose a method for detecting handwriting in sets of tabular document images that share a common form. This is accomplished leveraging previous work on aligning structured documents. These aligned documents are processed together as an image stack. First, the blank form common to the documents is generated using image averaging. Then, each image is compared individually to the blank form, and regions with a large difference are marked as handwriting. Results are pursued under the assumption that a good handwriting detection algorithm will have as few false positives as possible while maximizing recall. Proof of concept efforts are convincing, though a more indepth analysis remains to be done. Work to filter out false positives will be pursued.

1. INTRODUCTION

Images within a batch of census records often share in common the same form, but differ in handwritten content. We seek to extract the handwriting from the form and background. Our approach differs from previous efforts to separate handwriting from machine printed text, which commonly begin with connected component analysis, because we also must separate the handwriting from the form. Rather than using a bottom up approach with connected components, we leverage the redundancy between images to filter out the form, leaving only handwriting behind.

The ultimate goal in census indexing is to be able to convert both table headers and fields into text automatically, using both OCR and handwriting recognition[3]. For this to be possible, both machine print and handwriting have to be detected and isolated. We present a method for isolating handwriting in a batch of census records that share a common form[4].

Our specific contribution is an approach for handwriting detection in registered sets of tabular document images.

2. PREVIOUS WORK

Previous work in separating handwriting from printed characters generally begin with finding connected components to identify text. In [5], the components in a document are projected into an eigenspace, and then clustered. A simpler approach [2] uses the profiles of components to differentiate between handwriting and machine print. These approaches will not work in the case of census records, where the handwriting is usually connected to the form.

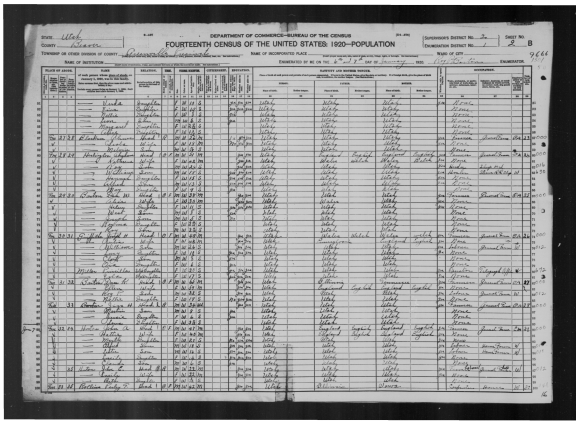

Figure 1: Form section of an image from the 1920 Utah census.

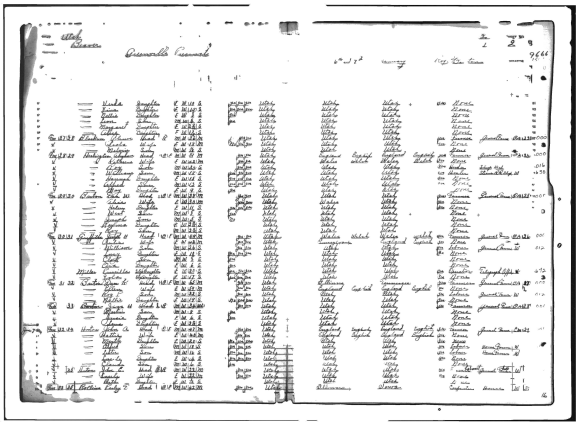

Figure 2: Extracted handwriting from 1920 Utah census.

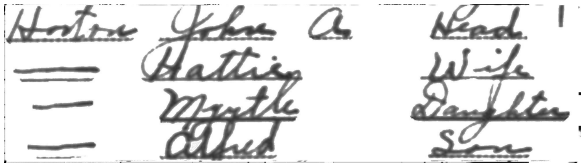

Figure 3: Close up view of handwriting.

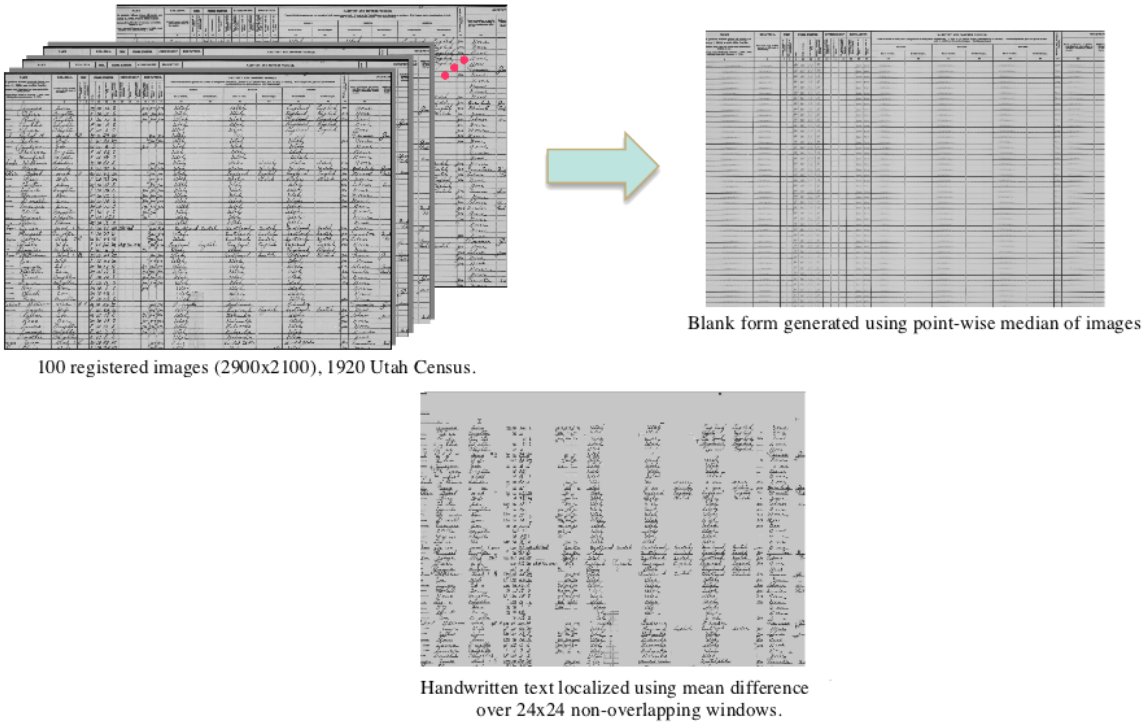

Figure 4: Sets of records (source images) are used to generate a blank form (template image). Then, the template is compared to each record in the set to isolate handwriting, generating new images where all non-handwriting pixels are made to match the background gray level.

While not directly related to handwriting detection, Luke Hutchison's work in registering images [1] is integral to our process, and is described in the next section. In addition, some previous work may be leveraged in our future work to filter out false positives.

3. IMAGE REGISTRATION

Image registration is "the process of finding the transformation that best maps one image to another"[1]. Using rotation, scale, translation, and/or shear transformations, common elements in registered images are aligned such that they will appear on the same place on each page. Image registration can be thought of as finding transformations that minimize the difference between images.

4. HANDWRITING DETECTION

We begin with a batch of records sharing the same form. Though the form is common to each image, images are not necessarily registered with respect to each other. We scale and register images using the Fourier-Mellin transform[1]. Once they are registered and scaled, the blank form is recovered by taking the median of the values across all images in the batch at each pixel location. Intuitively, this works because the background and form remain constant for each image, and are in the same location, but the handwriting changes from image to image. Having created the blank form, we isolate the handwriting in each image within the batch by thresholding ($t = 12$) the mean difference, d, of the image and the blank form over 24x24, non-overlapping windows. All pixels in a window with d < t are set to the median pixel value of the image. Thus, window pixels are preserved only if the content in the image is significantly different from the content in the blank form.

4.1 Generate Template Image

Having registered a sufficiently large set of images, it is possible to generate a template image, T, that contains only a blank form. The template image is created by taking the pointwise median across a set of registered images, as shown in equation 1.

Let $S_1 \ldots S_m$ be a set of registered documents, then

$$T(\mathbf{p}) \leftarrow Median(S_1(\mathbf{p}), S_2(\mathbf{p}), \ldots, S_m(\mathbf{p})) \qquad (1)$$

for all pixels, $\mathbf{p}[x, y]$.

While this method is very effective in recovering the blank form, there is a ghosting effect that can affect the accuracy of our algorithm (see Fig. 5). Currently, this is overcome by manually clearing the fields, but one focus of ongoing research is an automatic solution to this problem.

4.2 Detection Through Windowing

We present an algorithm for differentiating pixels in an image using overlapping windows and a voting scheme. An accumulator array, A, is used to store votes at all pixel locations. Handwriting is detected by synchronously passing a window pixel by pixel across both source and template images. If the pixels in the source vary enough from the pixels in the template, they are marked as handwriting. In our current implementation, the comparison between windows is a simple mean difference. If the difference exceeds

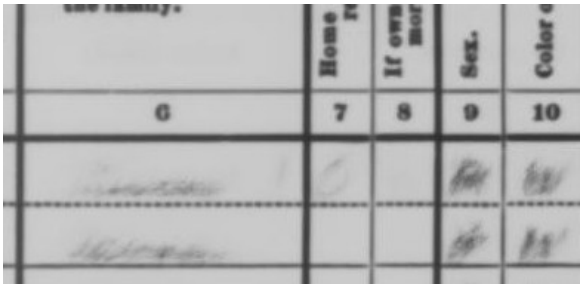

Figure 5: Handwriting ghosting in template image.

a threshold, the weight at each pixel location in A is increased. Conversely, if the difference is below the threshold, the weights are decreased in A for each pixel location in the window.

The accumulator array, A, is equal in size to the source image, and is initialized to 0. Handwriting votes increment values in A, and non-handwriting votes decrement values. When all windows have voted, the array A will contain both positive and negative values. A positive value at an $[x, y]$ location means that the pixel at location $[x, y]$ in the source image is handwriting. Negative values are background.

In our current implementation, once the accumulator array is generated, it is used in conjunction with the source image to create a new image, N, with the same dimensions. N is generated as follows.

Algorithm for Generating Resulting Image

foreach pixel $\mathbf{p}[x, y]$
  if $A(\mathbf{p}) > 0$
    $N(\mathbf{p}) = S(\mathbf{p})$
  else
    $N(\mathbf{p}) = background$

Currently, *background* is simply the average value of the entire source image. In future work, it would be better to choose a background level that matches the local region.

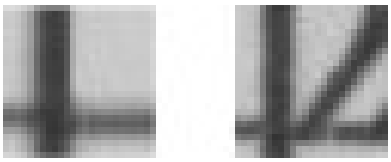

Figure 6: Example template (T) and source (S) windows.

Algorithmically, we show how an individual pixel is marked as either handwriting or background. Let $Q_1 \ldots Q_m, P_1 \ldots P_m$ denote all windows in S and T respectively that contain pixel $\mathbf{p}$.

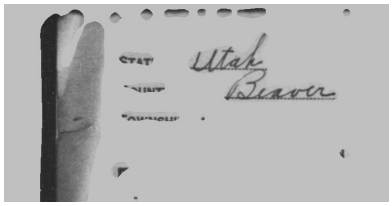

Figure 7: Borders of documents often have artifacts that are confused for handwriting because they don't exist in the template.

Algorithm for Filling and Reading the Accumulator

for n **in** $1 \ldots m$
  if $\left|\bar{Q_n} - \bar{P_n}\right| < t$
    $A(\mathbf{p}) := A(\mathbf{p}) + 1$
  else
    $A(\mathbf{p}) := A(\mathbf{p}) - 1$

if $A(\mathbf{p}) > 0$
  mark as handwriting

Notice that while we have simplified this algorithm for one pixel, in practice every pixel in each window receives a vote.

5. RESULTS

Preliminary results have yielded some promising images. Figure 3 shows the effectiveness of the algorithm in marking handwriting. However, there are situations where we do not do as well. In figure 7, we see that near the edges of the images, there can be artifacts unique to an image, that will appear very different than the template image. These will be indistinguishable from handwriting to our algorithm. Also, in figure 8, a piece of tape has been applied to the image. The region with the tape had a large difference compared to the template.

6. FUTURE WORK

While preliminary results are promising, there are still improvements that we are currently pursuing. Thus far, our attempt has been to leverage the statistics that are general to the set of images, like the median, and mean difference. We have also tried leveraging the pointwise variance of the set of documents, but found it too sensitive to noise, and not a strong differentiator between some locations of handwriting and the form. While stastical tools are impressive in their results given their simplicity, we have become less convinced that we will be able to use them exclusively to extract the handwriting out of these documents.

Because our images are axis aligned, finding higher order features, like discovering the geometry of the form, should not be too difficult. Using a matched filter on the horizontal and vertical profiles, we will extract the geometry of the form, and use the lines of the form as cues to extract the handwriting.

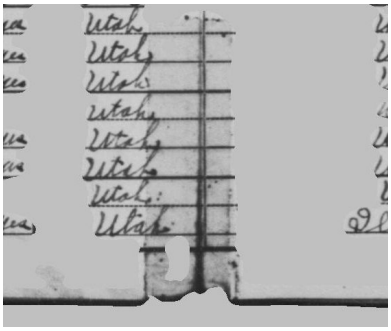

Figure 8: Some artifacts, like tape, will be unique to a particular image.

7. CONCLUSION

Tabular document images present interesting challenges in detecting handwriting. We presented a simple method for harnessing the similarity of multiple document images that share a common form. Results show that we are headed in the right direction, but more work remains to be done. For future work, we intend to exploit higher order features like the form geometry to help extract handwriting with greater accuracy.

8. REFERENCES

[1] L. Hutchison and W. Barrett. Fourier-mellin registration of line-delineated tabular document images. 8:87 – 110, June 2006.

[2] E. Kavallieratou and S. Stamatatos. Discrimination of machine-printed from handwritten text using simple structural characteristics. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR04)*.

[3] D. Kennard, W. Barrett, and T. Sederberg. Word warping for offline handwriting recognition. *ICDAR2011*, 1:1349–1353.

[4] H. Nielson and W. Barrett. Consensus-based table form recognition of low-quality historical documents. 8:183 – 200, June 2006.

[5] S. J. Pinson and W. A. Barrett. Connected component level discrimination of handwritten and machine-printed text using eigenfaces. *11th International Conference on Document Analysis and Recognition (ICDAR)*, 1:1394–1398, September 2011.