# Signature line detection in scanned documents

**Conference Paper** · September 2016

DOI: 10.1109/ICIP.2016.7532961



**8 authors**, including:

Osborn de Lima
Rochester Institute of Technology
**5** PUBLICATIONS **3** CITATIONS
SEE PROFILE

Eli Saber
Rochester Institute of Technology
**128** PUBLICATIONS **3,850** CITATIONS
SEE PROFILE

Mark Quentin Shaw
HP Inc.
**53** PUBLICATIONS **322** CITATIONS
SEE PROFILE

# SIGNATURE LINE DETECTION IN SCANNED DOCUMENTS

*Osborn de Lima*[1], *Shruty Janakiraman*[2], *Eli Saber*[1,2], *David C. Day*[3], *Peter Bauer*[3], *Mark Shaw*[3], *Roger Twede*[3], *Perry Lea*[3]

[1]Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology
[2]Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology
[3]HP Inc.

## ABSTRACT

In this paper, we present an algorithm to determine the signature line in scanned documents through a fast and computationally efficient technique. To that effect, the algorithm consists of three modules, namely pre-processing, candidate line detection and classification. Preprocessing in this case consists of hard thresholding, which is essential to line detection in the next module. The Hough transform for a single angle (horizontal) is utilized in the line detection module. Connected component analysis is utilized to merge detected Hough lines belonging to the same line on the document. Finally, the classification module attempts to classify each line as a signature line or not. This is accomplished using Histogram of a Gradient (HoG) features and a Euclidean distance measure for comparison. Prior data for this similarity comparison consists of images of target text in different font sizes and styles. The proposed algorithm showed favorable results in terms of precision and accuracy.

***Index Terms—*** Signature Detection, Document Identification, Object Recognition, Hough Transform, Horizontal Line Detection, Histogram of Gradients

## 1. INTRODUCTION

Signatures on documents are still very relevant and important in today's world for verification, identification and authorization. Even with the advent of the digital age, paper still plays an important role in business and personal transactions. However, the digitization of documents for improved storage and intelligent processing [1] has led to increased research in the field of Document Image Processing and Analysis.

Optical Character Recognition (OCR) is one such area of research that has been extensively studied, and other applications such as page layout classification and analysis are attributes that are either connected to the development of OCR engines or to the further advancement of document search and indexing. In [2], Doermann highlights the need for efficient indexing and retrieval systems for document images due to the increased reliance of large databases of such data due to their economic feasibility. A survey of document image processing and analysis and its various aspects can be found in [1], [2], [3] and [4].

Indexing, search and retrieval can done using a variety of attributes in document images. In [5], Zhu et al attempt to match logos from an existing database for efficient retrieval. In addition to detection and segmentation of logos using a cascade of classifiers across multiple image scales, they utilize rotational, translational and scale invariant shape descriptors and match them using developed shape similarity metrics. Ahmed et al in [6] utilize SURF features to detect signatures on documents and segment them out. They are able to accomplish this using SURF feature descriptors to distinguish handwritten text from machine printed text. A less trivial approach was adopted by Zhu et al in [7] in order to detect and match signatures for retrieval. They approached the problem of signature detection and segmentation by treating the signatures as symbols and leveraging the fact that they exhibit structural saliency. They employ a saliency model across multiple scales and were able to perform offline signature verification with great success.

The paper proposes to solve a different problem of detecting signature lines in document images that may or may not contain a signature. As such, prior work utilizing distinguishing features of signatures and handwritten text cannot be utilized. Also, computationally intensive techniques requiring large memory storage cannot be utilized due to the end goal of running the developed algorithm on a low memory embedded processor. This constraint disqualifies the use of OCR for searching desired text phrases. To this effect, a signature line detection scheme that operates under the premise of finding the lines and searching areas around them is proposed. Segmented words are treated as shapes and Histogram of a Gradient (HoG) features [8] are extracted. A simple feature similarity measure is utilized and a decision threshold is established to classify a line as one of interest or not. Extensive prior data is used to alleviate the problems of scale and style.

The remainder of the paper is organized as follows. Section 2 briefly highlights the proposed algorithm and its

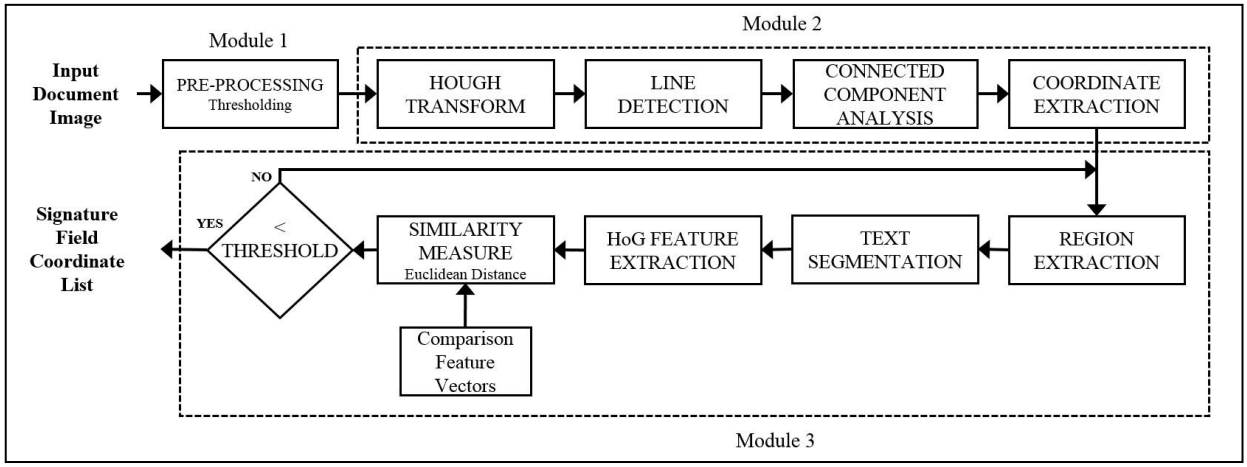


**Fig. 1:** Flowchart of the proposed algorithm

constituent modules. Results are shown and discussed in Section 3 and conclusions are drawn in Section 4.

## 2. PROPOSED ALGORITHM

An overview of the proposed technique is shown in Fig. 1. It consists of three main modules, namely pre-processing, candidate line detection and classification. Pre-processing could consists of noise removal, enhancement in addition to thresholding. The subsequent module utilizes the Hough Transform along with connected component analysis to merge multiple redundant lines. Finally, from those multiple candidate lines, regions around each of them are extracted and progressively text is segmented. HoG features for words within a defined aspect ratio are extracted and compared with prior data. The smallest distance metric among all the segmented words is saved for each candidate line. The candidate lines that have a similarity metric that is less than a pre-defined threshold are classified as signature lines.

### 2.1 Pre-processing

Pre-Processing is shown as a separate module even though for our test cases it only consists of a single step, namely thresholding. It is fundamental to the line detection process and the level at which this occurs effects the line detection process in terms of length of each line found and number of lines found. Experimentally, a threshold of 0.65 (with intensity levels ranging from 0 to 1) was chosen which in turn produced the most satisfactory results. The resulting document image has white pixels on a black background.

### 2.2 Candidate Line Detection

The Hough Transform is a common technique used in image processing for the global detection of lines of different angles in images. The fundamental concept behind it is expressing the *xy* plane where the image lies in the *parameter space*. Since it is horizontal lines we are seeking, Eqn. 1 is evaluated for all *x* and *y* coordinates that are 1 in the binary image with $\theta = \pi/2$.

$$x\cos\theta + y\sin\theta = \rho \tag{1}$$

The Hough Transform and Hough Line detection [9] process results in horizontal lines of at least 40 pixels in length to be found. This is termed as a *HoughLines* map. However, multiple lines are found for a single straight line in the document. This is due to the fact that at a pixel level for a scanned document, a single line will be multiple pixels thick. Additionally, due to the skew associated with the image acquisition process, a single line in the document is made up of several disjointed horizontal Hough Lines. In order to reduce the number of redundant lines and consequently the number of search regions, a merging process is required.

Connected Component Analysis with an eight connectivity basis is applied on the binary image wherein all the pixels part of the lines determined in the prior step are set to 1. The eight connectivity basis allows for a pixel under consideration to be labelled as belonging to a certain group, if any of the four pixels to the top, bottom, right, or left.

### 2.3 Classification

In order to determine if the candidate line is indeed a signature line, the region around the line needs to be examined and searched for key words such as *Signature*, *Sign* etc. in various fonts and sizes. The first step in that process is extracting the region for processing for each candidate line separately.

#### 2.3.1 Region Extraction

Region Extraction doesn't only involve removing a fixed region around the line, but prior to that, examining presence of content below the line and to the left of the line. This is done by looking at a small region and checking for the width of content. Since the black line is off no interest in the current stage of processing, pixels containing the line are set

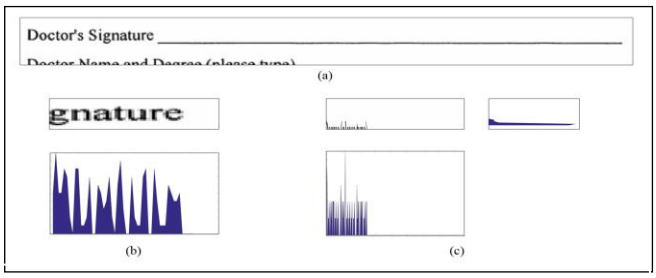


**Fig. 2:** Region Extraction: Identifying region around the line to extract

to high intensity using the *HoughLines* map obtained from the Line Detection stage in the previous module. Vertical Projections on the binary patch with white text pixels on a black background, gives us information of the width of text. If the projection profile has at least five pixels of content in there at any point, it implies that the region is of interest. It should be highlighted that when examining the region to the bottom of the line, an extra parameter of where the content lies i.e. not too far below the line, is taken into consideration. The aforementioned criterion allow us to extract the right region. If content exists both below and to the left of the line, the entire region is extracted. In Fig. 2, the line region (Fig. 2(a)) to the left of the line is searched (Fig. 2(b)) and returns a true condition since, the projection profile has one region of more than five pixels width. The region below the line has two criterion. Since the vertical projection profile shown below Fig. 2(c) satisfies the five pixel width condition, the region would be extracted as well. However, the horizontal projection profile to the right of Fig. 2(c) shows that the content is less than 10 pixels high, implying it is most likely not belonging to the region around the line in consideration.

2.3.2 Text Segmentation

A simple procedure to isolate text and analyze it for the intended purpose is achieved through multiple passes of Horizontal and Vertical Projections. First, a horizontal projection on the extracted region separates it into horizontal strips. Then for each horizontal strip, vertical projections are used to segment individual words and symbols. The parameters for the projections are accordingly adjusted so as to group words together and avoid segmenting individual letters. Also, it should be noted prior to entering the classifier, a final horizontal segmentation occurs in order to compensate for skewed text caused due to the scanning process. Prior to entering the HoG Euclidean Distance Classifier for comparison, an aspect ratio test is implemented. The aspect ratio in this case is defined as the ratio of the width of the image to height of the image in pixels. Only sample texts with aspect ratio between 1.6 and 6.7 are sent to the classifier. This allows long words, or small words, single letters or graphics that are not of interest to be avoided and not classified, thus saving on processing time and resources.

2.3.3 Similarity Measure of HoG Feature Vectors.

Histogram of Gradients (HoG) is a global feature descriptor developed by Dalal and Triggs [8] for Human Detection in images. The global nature implies that each feature vector generated for a given image resolution is of the same size. In a general sense, HoG feature vectors intend to utilize the wealth of information that is representative of objects in images, encoded in the Gradient Magnitude and Direction.

In the proposed algorithm, HoG features are evaluated on 8 x 8 *Blocks* with a *Cell* size of 4 x 4. This results in 4 cells in a block. There is a 50 % overlap between blocks. Block size and Cell size are intimately related to the amount of detail you are intending to capture. Since we are looking at text and associated detail is on a very small and local level a small block size and cell size are used. However, reducing the block size to a very small value can result in a very large feature vector.

In order to achieve an effective comparison through the Euclidean Distance measure between feature vectors, they need to have the same dimensionality. In the case of a global feature descriptor like HoG, the images have be of the same size. Therefore, in both the actual processing and when generating prior data for comparison, the images are resized to a size of 100 x 30 [W x H]. A fast nearest neighbor interpolation technique is employed since the level of resizing is not significant, and vital information will not be lost.

The HoG features of the sample in consideration is compared with each of the prior feature vectors. The smallest distance is stored for each case and for each candidate line. Finally, after all the candidate lines have been processed, a threshold of 65 is set, implying the squared distances below that value indeed indicate a signature line. The value for the threshold was determined experimentally. The resulting coordinate list can be utilized for highlighting, zooming into those regions or further processing of said regions.

2.3.4 Comparison Feature Vectors

Since the classifier employed here (similarity measure, followed by a decision threshold) is trivial in nature and, in being so provides us obvious computational advantages, prior data is used for explicit comparison to the data being examined. In an offline process, HoG Feature Vectors are extracted for each prior word. For each font type, variants of the word signature along with associated colons as is common in many application forms is typed in a Word Processor in seven fonts sizes (12-24 point). This allows for some degree of scale invariance. That being said, the algorithm assumes a few degrees of rotation and skew, but is not rotationally invariant. The prior comparison data is scanned so as to mimic the same image acquisition process and segmented using a similar procedure as highlighted in the prior section. HoG features are extracted and stored in a data array. HoG feature parameters such as Cell Size and

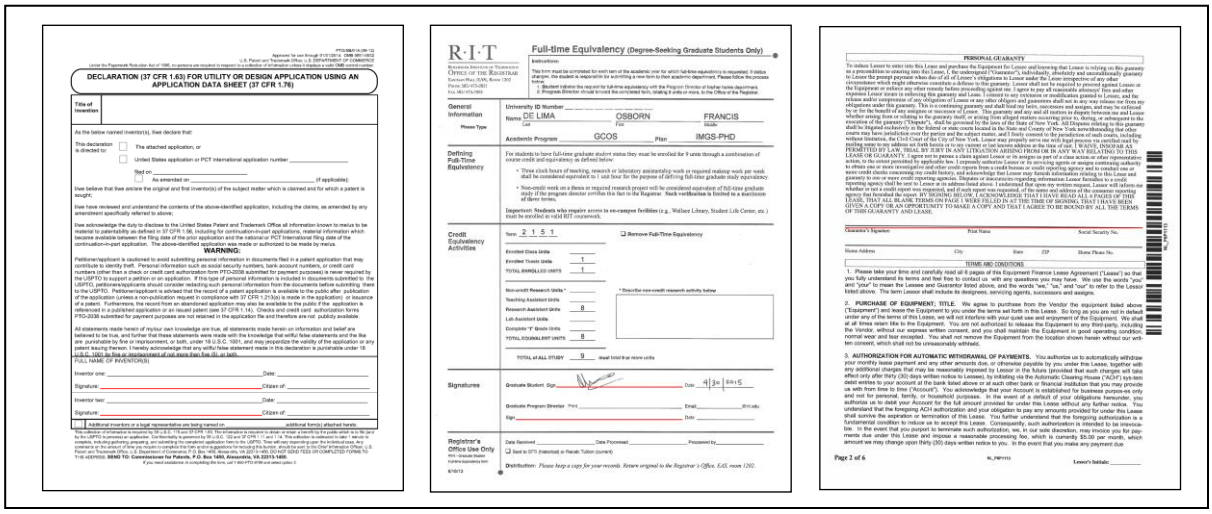

**Fig. 3:** Results on a few of the documents in the Test Suite

Block Size are kept the same as in the actual algorithm to get feature vectors of the same length as will be extracted for sample data during processing for signature line detection. In our work, we collected 133 feature vectors, corresponding to 133 instances of the word *signature, sign*, etc.

## 3. RESULTS AND DISCUSSION

The proposed algorithm was implemented in MATLAB and C and was evaluated based on precision and recall metrics. Since there has been no prior work in the field of detecting signature lines, an appropriate benchmark couldn't be performed.

The algorithm was tested on a suite of 38 documents consisting mostly of various types of application forms. It is important to note that one of the assumptions made during this research, was the presence of a line for a signature. Also, during initial development, the primary target device was a scanner, so the test documents were acquired by scanning them at a resolution of 300 dpi in grayscale.

A few examples of the results are shown in Fig. 3. The documents shown have the words *Sign*, *Signature* and *Signed* denoting the signature line. For the purpose of visualization, the signature line is underlined, however, the algorithm returns the coordinates of the signature line/lines, which can be used to zoom into locations for a user to sign on a tablet or display.

The results show that the algorithm achieved a **Precision** of **95.23%** and **Recall** of **98.36%**. In this case. Precision is defined as the fraction of returned signature lines that are indeed signature lines, thereby accounting for false positives. Recall is defined as the fraction of signature lines returned out of the total number of signature lines in the test document set, which accounts for false negatives. In terms of speed, the average time taken on a desktop computer (8 Cores, 3.50 GHz, 16 GB RAM) was around **5.86 seconds** in MATLAB and **5.32 seconds** in C. However, it should be noted that the C implementation performed much faster for documents with a large number of lines.

False positives were deemed more acceptable than false negatives. Out of a total of 61 potential signature lines in the 38 documents, only one of them was missed. Two determined failure modes of the algorithm are dotted lines and gridded documents. In the case of dotted lines, they are not detected currently by the line detection module, so they are not even considered for classification. In the case of gridded documents, the failure to detect signature lines arises from the manner in which the search is done around a line. The premise of the key word being to the left or below the line fails in such cases.

Future research could include improvements to the manner in which the region around a line is searched, accounting for gridded documents and reducing false positives for documents that have lines that are in close proximity to others. Also, additional shape descriptors can be evaluated and compared with HoG to study its performance.

## 4. CONCLUSIONS

This paper describes a signature line detection process for scanned document images using the hough transform for line detection and HoG features to classify candidate lines. A similarity measure to compare candidate words with prior feature vectors consisting of the word signature and its variants in multiple sizes and styles. The algorithm showed promising results on a limited test document suite and due to its low complexity can be implemented on a single chip processor for embedded computing in commercial devices. In addition, it has added scope in determining if a signature has been placed in all required places for important legally binding documents.

## 5. REFRENCES

[1] S. Akram, M. Dar and A. Quyoum, "Document Image Processing - A Review", International Journal of Computer Applications, vol. 10, no. 5, pp. 35-40, 2010.
[2] D. Doermann, "The Indexing and Retrieval of Document Images: A Survey", Computer Vision and Image Understanding, vol. 70, no. 3, pp. 287-298, 1998.
[3] U. D. Dixit and S. M.S, "A Survey on Document Image Analysis and Retrieval System", IJCI, vol. 4, no. 2, pp. 259-270, 2015.
[4] L. O'Gorman and R. Kasturi, Document image analysis. Los Alamitos, Calif. [u.a.]: IEEE Computer Society Press, 1997.
[5] G. Zhu and D. Doermann, "Logo Matching for Document Image Retrieval", 2009 10th International Conference on Document Analysis and Recognition, 2009.
[6] S. Ahmed, M. Malik, M. Liwicki and A. Dengel, "Signature Segmentation from Document Images", 2012 International Conference on Frontiers in Handwriting Recognition, 2012.
[7] G. Zhu, Y. Zheng, D. Doermann and S. Jaeger, "Signature Detection and Matching for Document Image Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 11, pp. 2015-2031, 2009.
[8] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).
[9] Mathworks.com, "Extract line segments based on Hough transform - MATLAB houghlines", 2016. [Online]. Available:http://www.mathworks.com/help/images/ref/houghlines.html?requestedDomain=www.mathworks.com. [Accessed: 01- Feb- 2016].