# Detecting text in floor maps using Histogram of Oriented Gradients

**Conference Paper** *in* Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on · October 2013

DOI: 10.1109/ICASSP.2013.6637990

Qiongjie Tian
Arizona State University

# DETECTING TEXT IN FLOOR MAPS USING HISTOGRAM OF ORIENTED GRADIENTS

*Hima Bindu Maguluri, Qiongjie Tian and Baoxin Li*

Computer Science and Engineering, Arizona State University, Tempe, USA

## ABSTRACT

Automatic detection of text labels in maps is essential for applications requiring automatic map understanding. This task is challenging due to factors such as varying font size and style, slanted words/phrases, and interfering graphics that are similar to text. This paper presents an approach for text detection in indoor floor maps. We exploit the difference in spatial frequency of edge orientations between text and non-text regions through Histogram of Oriented Gradients (HOG) features, and design a gradient-filtered Support Vector Machine (SVM) classifier based on such features. Special care was taken in conditioning the data for proper training of the classifier. The proposed approach was evaluated on a data set that had been collected and manually labeled. Experimental results show that the proposed method attained improved performance, outperforming a couple of reference methods/systems.

***Index Terms***— Text Detection, Histogram of Oriented Gradients, Support Vector Machine.

## 1. INTRODUCTION

There are many applications that require automatic map understanding, a key task of which is accurate text detection from maps. Examples include querying map images by keywords and making maps accessible to the visually impaired. In the field of computer vision and image understanding, many methods have been proposed for text detection in images (e.g., [1] [2] [3]). While advances have been made, typical existing approaches do not deliver the desired accuracy demanded by practical applications. Further, many approaches were reported in different contexts and were evaluated with different data sets, making it difficult to perform comparative assessment in understanding the effectiveness of the methods.

In this work, we focus on the task of detecting text from floor maps, with an ultimate goal of assisting map understanding (and thus high accuracy in detection is the key). Factors such as varying input resolution, diverse font size and style, and slanted text are typical of floor maps from different sources, and they often baffle existing approaches that rely on a small training set or a set of fixed rules. We explore the spatial frequency of edge orientations using HOG features, and design a gradient-filtered SVM classifier based on such features. Training the classifier was facilitated by conditioning the data to provide better labels. Further, recognizing that there is no standardized data set for comparing the performance of different approaches, we collected and manually labeled a data set for our experiments. The experimental results show that the proposed method attained improved performance, outperforming a couple of reference methods/systems. The data set is available for other researchers to evaluate and compare their method.

## 2. RELATED WORK

Most of the recent work on text detection has been focused on natural images [3] [4] [5] [6] [7] [8] [9]. Such algorithms are in general not tuned for text detection in floor maps, which exhibit different properties from natural images. For example, in a floor map, there are often graphics that are in many ways similar to texts. There are also some methods for text detection in graphic background. In [10], extraction of text regions in multiple colors and complex backgrounds was discussed. In [11], an approach to separate text from the graphics was presented, focusing on recovering text that has been overlapped by graphics. In [12], a method was proposed to detect text regions by thresholding local frequency features. Unfortunately, while these methods have their potential strength and weakness, there is no systematic comparison of their accuracy based on a common test set, and thus little can be concluded on their performance in general. For example, it is difficult to expect that the technique of thresholding frequency features can differentiate text from other text-like graphics, which occur frequently in floor maps.

We propose to use edge orientation information as the key feature since texts have distinct shapes when compared with general graphics. This is a proper feature especially since the floor maps are largely binary in nature. This is achieved by computing the HOG features ([13]). Further, to avoid setting any hard threshold, which is difficult to do, we learn a SVM classifier based on the features. Also recognizing the potential of the gradient distribution in distinguishing texts from their surroundings, the SVM classifier is modulated by a gradient-

The work was supported in part by a grant (#0845469) from the National Science Foundation. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

based filter. To ensure proper training, we also propose a few data conditioning techniques.

## 3. THE PROPOSED METHOD

The proposed method consists of three major processing steps, which are detailed in the following sub-sections.

### 3.1. Extraction of HOG features

As a preprocessing step, a floor map image is first converted into a grayscale version and then smoothened by using a Gaussian filter to avoid excessive noise in gradient computation. A small Gaussian kernel is used to preserve major edges. The magnitude and orientation of the gradient at each pixel are obtained by convolving the smoothened image with derivative filters along the horizontal and vertical directions. In theory the gradient magnitudes in homogeneous regions of the image are zero. In practice, due to factors like compression artifacts, even pixels of visually homogeneous regions may have small non-zero gradient magnitude, and the corresponding gradient orientation is not useful for text detection. To alleviate this effect, a constant threshold $k$ has been applied to the gradient magnitude, as shown in Equation 1, where $M$ is the gradient map (magnitude) of the given image. Also, the text regions become more obvious after this thresholding step as shown in Figure 1.

$$M_{i,j} = \begin{cases} 0 & , if\ M_{i,j} \leq k \\ M_{i,j} & , if\ M_{i,j} > k \end{cases}, \quad \forall i, j \tag{1}$$

For each pixel, a $n \times n$ window centred at the given pixel is considered to construct a $d$ dimensional HOG feature. The height of each bin in the histogram is calculated as the number of pixels from the $n \times n$ window with non-zero gradient magnitude and with orientations that fall in the range of the particular bin. The number of bins $d$ has been chosen in such a way to capture the orientation information in terms of smaller ranges because text regions have wide variety of angles. This makes the histograms suitable for differentiation between text and non-text regions. Figure 2 illustrates examples of the histograms corresponding to text and non-text regions respectively.

### 3.2. Data conditioning

The histogram features corresponding to text and non-text pixels are labeled as $+1$ and $-1$ respectively by selecting rectangular boundaries that surround the text. An example of ground truth is shown in Figure 3. Due to variations in height and shape of the text characters, sometimes pixels that are labeled as $+1$ and located inside the rectangle surrounding some texts can have very small number of edge orientations in its neighborhood and thus have features similar to non-text regions. Examples include pixels lying in the borders of rectangles or pixels in between letters. To accurately train a subsequent classifier, we introduce a data conditioning step to identify and relabel such pixels, thus providing better labels for training. This is formally done by Equation 2,

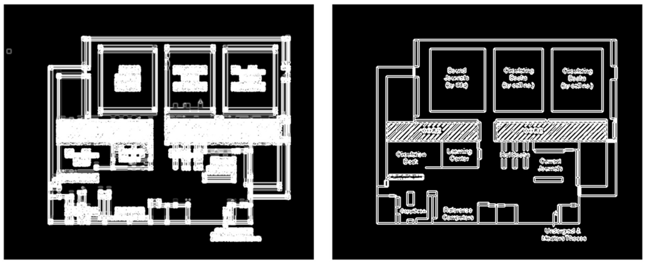


**Fig. 1**. Visualization of magnitude of gradients before (left) and after (right) thresholding.

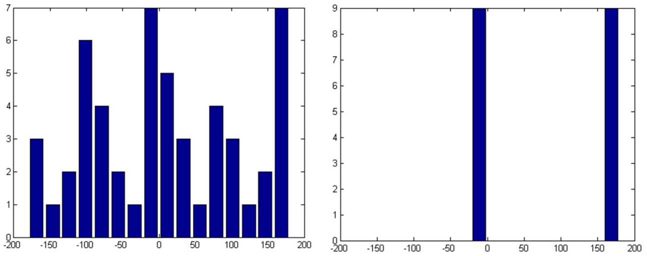


**Fig. 2**. Examples of HOG features corresponding to text (left) and non-text regions (right).

$$\begin{aligned} X_{i,j} &= \begin{cases} 0 & , if\ M_p(i,j) = 0 \\ 1 & , if\ M_p(i,j) > 0 \end{cases}, \quad \forall i, j \\ N_p &= \sum_{i=1}^{n} \sum_{j=1}^{n} X_{i,j} \\ C_l &= \begin{cases} -1 & , if\ N_p < t \\ l & , if\ N_p \geq t \end{cases} \end{aligned} \tag{2}$$

where $M_p$ represents thresholded gradient magnitude values of a $n \times n$ patch centered at the given pixel, $N_p$ represents the total number of pixels with non-zero gradient values in the patch, and $t$ is a threshold. $l$ and $C_l$ are ground truth and conditioned labels respectively, as visualized in Figure 3, where it can be seen that the conditioned labels represent text pixels more accurately than the initial ground truth.

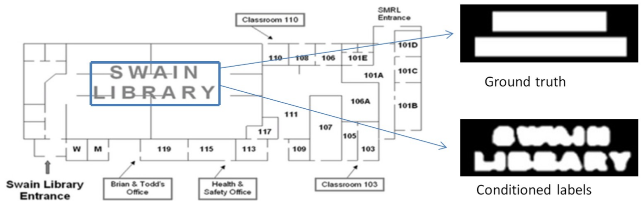


**Fig. 3**. Examples of ground truth and conditioned labels.

### 3.3. Gradient-filtered SVM classification

With the labeled features, an SVM classifier is trained and used for predicting text regions in test images. Example of the SVM-predicted result can be seen in Figure 4. It is observed that the detected regions are not clean enough and they include many surrounding non-text pixels. This gives rise to many false detections and poor localization with each of output rectangle including too many words along with surrounding noise. As the gradient distribution has the potential to detect accurate boundaries, the SVM-predicted output is filtered with non-zero gradient magnitude values as shown in Equation 3,

$$H_{i,j} = \begin{cases} 0 & ,if\ M_{i,j} = 0 \\ 1 & ,if\ Mi,j > 0 \end{cases}, \ \forall i,j \qquad (3)$$
$$Y_{i,j} = P_{i,j} * H_{i,j}, \ \forall i,j$$

where $M$ represents the thresholded gradient image from Equation 1, $P$ the SVM-predicted output, $H$ the gradient-based filter and $Y$ the filtered output. This eliminates many false detections and enhances localization by providing tighter boxes surrounding texts.

To further eliminate any false positives or noise resulting from lines or large connected components, the output $Y$ is post-processed with filters obtained from a line detection module and detection of large connected components.

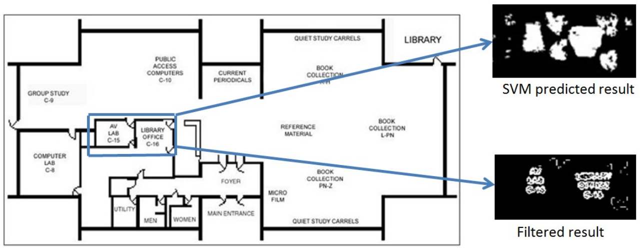


**Fig. 4**. Example of SVM-predicted result and gradient-filtered result.

## 4. EXPERIMENT DESIGN AND RESULTS

There is strong need for standardized data set of maps to ensure appropriate comparison of various methods on a common test set. We created a data set, by collecting floor maps of 30 libraries and manually marking the ground truth. The data set has been created in such a way that it includes diverse images with different variations in structure of the building, image resolution, average text height etc. This dataset is available at `http://www.public.asu.edu/~bli24/icassp2013.html` for any interested researcher to use. The current version include only library floor maps and the dataset may be updated with other types of floor maps in the future.

| $Results from:$ | Precision | Recall | F |
|---|---|---|---|
| Our algorithm | 85.8% | 57.9% | 69.2% |
| Algorithm of [12] | 45.7% | 90.6% | 60.8% |

**Table 1**. Comparison of pixel level accuracies from our algorithm and from [12].

The proposed method has been evaluated on our data set and compared with results from the text detection method used in [12]. We chose the method from [12] for comparison since it was designed for a very similar problem (text detection from on-line maps). Out of the 30 images, 19 images were used for training the SVM and the remaining 11 were used for testing. Learning SVM model and classification have been achieved using LIBSVM tool [14]. Window size $n$ was chosen as 9, number of bins $d$ as 16 and the constants $k$ and $t$ as 20 and 15 respectively. Experimental results are presented at two levels: pixel level and word level.

### 4.1. Pixel-level evaluation

For each image, the precision and recall values were calculated in terms of number of pixels and then, averages of the precision and recall values over all the 11 images were calculated. Comparison of the values of precision, recall and the standard F1 score, $f$ from the proposed method and the method of [12] is shown in Table 1. It can be seen that our algorithm has higher precision value. Though the numerical value of recall is low for our algorithm, Figure 5 shows that the output obtained by our algorithm is cleaner and more accurate (both in terms of detecting text regions and eliminating false detections).

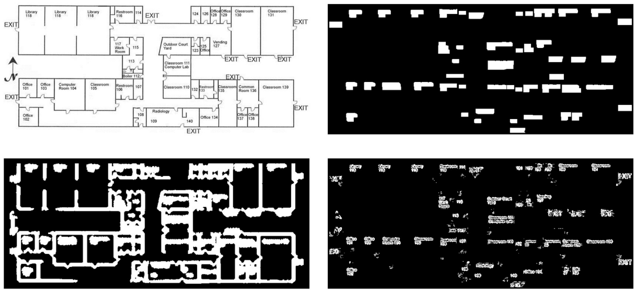


**Fig. 5**. Input (top left), ground truth (top right), detected result from [12] (bottom left) and our algorithm (bottom right).

The reason for the recall value being low is that the ground truth is a solid rectangle with its inside filled but we have refined our output by gradient filtering to remove unwanted pixels as explained in Section 3.3. The decrease in recall value when calculated in terms of number of pixels is due to the

| $Resultsfrom:$ | Precision | Recall |
|---|---|---|
| Our algorithm | 67.4% | 88.8% |
| Algorithm used in [12] | 50.0% | 61.0% |

**Table 2**. Comparison of word level accuracies from our algorithm and [12].

difference in the nature of the output mask we obtained (Instead of solid filled output, we obtained clean output which enhances localization) and results in Section 4.2 show that this difference in nature of the mask achieves better localization and thus higher word level accuracy. The nature of our output mask is shown as Filtered result in Figure 4 and it can be noticed that the filtered result has clear boundaries. This demonstrates that the low recall value in Table 1 actually indicates that our algorithm performs better than [12] by generating a cleaner output for better localization. We also plotted the ROC curve and observed that the AUC (Area Under Curve) was 0.942.

### 4.2. Word-level evaluation

To obtain meaningful recognition results on detected text regions, accurate localization of text is essential. The detected boxes should be tight enough, so that they do not include surrounding graphics. For this purpose, we calculate the word-level accuracy to evaluate the detection and localization. We use coordinates of the ground truth rectangles to evaluate the bounding boxes obtained by our algorithm and we compare with the results from [12].

For each ground truth text box, a recall value is calculated as the ratio of the overlapping area between the ground truth box and the detected text boxes to the area of the ground truth text box. For each detected text box, a precision value is calculated as the ratio of overlapping area between the detected box and the ground truth boxes to the area of the detected text box. The average of the precision and recall values over all the text boxes from the 11 test images were calculated. Comparison of the precision and recall values at word level is shown in Table 2 and it can be seen that our method has significantly higher precision and recall values.

These results support our explanation in Section 4.1 that lower recall value in Table 1 is due to cleaner mask obtained by removing unwanted pixels and thus our algorithm enhances localization and achieves higher word level accuracy. Figure 6 shows an example of the detected text boxes from [12] and from our algorithm.

### 4.3. Comparison with OMNIPAGE

We now present results from a commercial OCR software, OMNIPAGE 2007. We observed the results from OMNIPAGE on all the test images and in most of the cases, the software failed to give reasonable performance. As an example, Figure 7 shows the screen shot of results obtained when a floor map is given as input to OMNIPAGE. It can be seen that the output text file has very few meaningful words. The brown polygonal regions marked on the input image show the detected regions. The software could not detect accurate boundaries of text regions from the input image and even in the detected regions, it could not return meaningful words due to the presence of many interfering lines and graphics. It was also observed that when the cropped regions of detected text boxes obtained from our algorithm were given as input, the number of meaningful words were much higher. This shows the usefulness of an accurate text detection and localization method in recognizing texts for automatic map understanding.

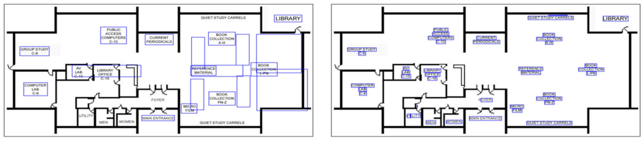
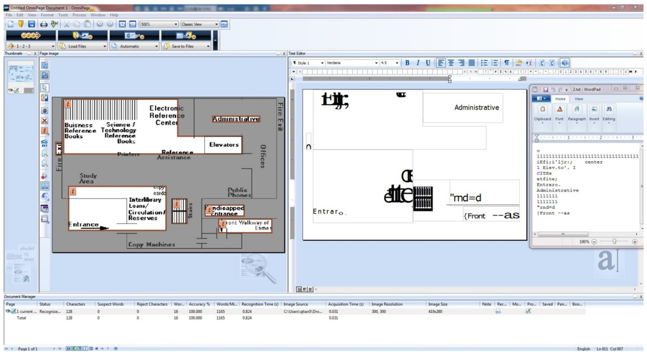


**Fig. 6**. Localized text boxes from [12] (left) and our algorithm (right).



```
o
1111111111111111111111111111111
iEfi;i'ljr:;    center
1 Eiev.to'. I
CITSs
etfite;
Entraro.
Administrative
1111111
1111111
"rnd=d
{Front --as
```

**Fig. 7**. OMNIPAGE results for an input floor map.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we reported an approach to text detection in floor maps. We analyzed the challenges involved and the deficiencies of typical existing approaches. Then we presented our method using edge orientation information in the form of HOG features and a gradient-filtered SVM classifier. Experimental results demonstrate the usefulness of selected features and robustness of the proposed method even in handling slanted text in low-resolution images. For future work, we plan to extend the test dataset and also incorporate OCR feed back into our system to eliminate false detection and recover missing regions of partially detected text boxes.

## 6. REFERENCES

[1] Chen, D. and Bourlard, H. and Thiran, J.P., "Text identification in complex background using svm," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, vol., pp., II–621.

[2] Pan, Y.F. and Hou, X. and Liu, C.L., "A hybrid approach to detect and localize texts in natural scene images," *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 800–813, 2011.

[3] Minetto, R. and Thome, N. and Cord, M. and Stolfi, J. and Précioso, F. and Guyomard, J. and Leite, NJ, "Text detection and recognition in urban scenes," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, pp., 227–234.

[4] Escalera, S. and Baró, X. and Vitrià, J. and Radeva, P., "Text detection in urban scenes," in *Proc. Conf. on Artificial Intelligence Research and Development*, pp., 35–44.

[5] Neumann, L. and Matas, J., "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp., 3538–3545.

[6] Epshtein, B. and Ofek, E. and Wexler, Y., "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp., 2963–2970.

[7] Chen, X. and Yuille, A.L., "Detecting and reading text in natural scenes," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, vol., pp., II–366.

[8] Ezaki, N. and Bulacu, M. and Schomaker, L., "Text detection from natural scene images: towards a system for visually impaired persons," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, vol., pp., 683–686.

[9] Ye, Q. and Jiao, J. and Huang, J. and Yu, H., "Text detection and restoration in natural scene images," *Journal of Visual Communication and Image Representation*, vol. 18, no. 6, pp. 504–513, 2007.

[10] Tian, Y.L. and Yi, C. and Arditi, A., "Improving computer vision-based indoor wayfinding for blind persons with context information," *Computers Helping People with Special Needs*, vol. 18, no. 6, pp. 255–262, 2010.

[11] Cao, R. and Tan, C., "Text/graphics separation in maps," *Graphics Recognition Algorithms and Applications*, vol. 18, no. 6, pp. 167–177, 2002.

[12] Wang, Z. and Li, B. and Hedgpeth, T. and Haven, T., "Instant tactile-audio map: enabling access to digital maps for people with visual impairment," in *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*. ACM, pp., 43–50.

[13] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, vol., pp., 886–893.

[14] Chang, C.C. and Lin, C.J., "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.