# Automatic Text Location in Images and Video Frames

Anil K. Jain and Bin Yu

Dept. of Computer Science, Michigan State University
East Lansing, MI 48824, USA

## Abstract

*Automatic text location (without character recognition capabilities) deals with extracting image regions that just contain text. The images of these regions can then be fed to an optical character recognition module or highlighted for users. This is very useful in a number of applications such as database indexing and converting paper documents to their electronic versions. The performance of our automatic text location algorithm is shown in several applications. Compared with some traditional text location methods, our method has the following advantages: (i) low computational cost; (ii) robust to font size; and (iii) high accuracy.*

## 1. Introduction

Generally, there are two goals of automatic text processing. (i) converting text from paper documents to their electronic versions [3], and (ii) finding useful information about the documents (e.g., image, video, paper document) which contain text. It is the second application which plays an important role in Web search, color image indexing, database organization, automatic annotation and video indexing. An automatic text location algorithm (the first step in automatic or semi-automatic text reading) extracts regions which just contain text from various text carriers without recognizing characters contained in the text. The expected variations of text in terms of character font, size and style, orientation, alignment, texture and color embedded in low contrast and complex background images make the problem of automatic text location very difficult. Furthermore, a high speed of text location is desired in most applications.

Several approaches to text location have been proposed for page segmentation [3], address block location [10, 7], form dropout [9] and graphics image processing [1]. These approaches are based on the following two methods. The first method regards text as a textured region and uses well-known methods of texture analysis [6] such as Gabor filtering [2] and spatial variance [11] to automatically locate text regions. This approach is sensitive to character font size and style. Further, this method is generally time-consuming and can not always accurately give text's location which may reduce OCR's performance. The second method of text location uses connected component analysis [9, 10, 3, 8]. This method, which has a higher processing speed and localization accuracy, however, is applicable to only binary images. We localize text through multivalued image decomposition. The proposed method has been applied to the problem of locating text in a number of different domains, including classified advertisements, embedded text in synthetic Web images, color images and video frames.

## 2. Text Location Algorithm

The most important requirements for our text location applications are: (i) high speed, and (ii) location of only important text in the input image. Usually, the larger the font size, the more important is the text. The text which is very small in size cannot be recognized easily by OCR engines anyway [12]. Generally, the important text in images appear mainly in the horizontal direction. Therefore, our method tries to extract only horizontal text of relatively large size. However, our algorithm can be easily modified to extract text in other directions as well. Because some non-text objects can be subsequently rejected by an OCR module, we minimize the probability of missing text (false dismissal) at the cost of increasing the probability of false alarm. Fig. 1 gives an overview of the proposed system. The input can be a binary image, a synthetic Web image, a color image or a video frame. After color reduction (bit dropping and color clustering) and multivalued image decomposition, the input image is decomposed into multiple foreground images. Individual foreground images go through the same processing steps, so the connected component analysis and text identification modules can be implemented in parallel on a multiprocessor system to increase the processing speed. Finally, the outputs of all the channels are fused to locate the text in the input image. Text location is represented as the coordinates of the bounding box surrounding the text. Details of our algorithm are provided in [4].

## 3. Experimental Results

The proposed system for automatic text location has been tested on a number of binary images, pseudo-color images, color images and video frames. Since different appli-

cations need different heuristics, the modules and parameters used in the algorithm shown in Fig. 1 change accordingly. Table 1 lists the performance of our system. We compute the accuracy for advertisement images by manually counting the number of correctly located characters. The accuracies for other images are subjectively computed based on the number of correctly located important text regions in the image. The accuracy for color images is the lowest because of the complexity of the background in these images. The processing time is reported for a Sun UltraSPARC I system (167 MHz) with a 64MB memory. More details of our experiments for different text carriers are explained in the following sub-sections.

### 3.1. Advertisement Images

The test images were scanned from a newspaper at 150 dpi. One text location result is shown in Fig. 3. The line of white blocks in the upper part of Fig. 3 is located as a text because the blocks are regularly arranged in terms of size and alignment. It should be rejected by an OCR module.

### 3.2. Web Images

Several Web images were down-loaded through the Internet. One result of text location is shown in Fig. 2. Note that Fig. 2 contains a title with Chinese characters which has been correctly located.

### 3.3. Scanned Color Images

Several color images were scanned at 50 dpi from magazine and book covers. One typical result of our algorithm is shown in Fig. 4. Most of the important text with sufficiently large font size has been successfully located. Some characters with small size are missed, but they are probably not important for image indexing.

### 3.4. Video Frames

Test video frames were selected from eight different videos covering news, sports, advertisement, movie, weather report and camera monitor events. The resolution of these video frames ranges from $160 \times 120$ to $720 \times 486$. The results in Fig. 5 show the performance of our algorithm on video frames with as low resolution as $160 \times 120$, where text font, size, color and contrast change within a large range. Note that it is often not easy even for humans to locate all the text in a video frame due to its low resolution. Non-caption text is more difficult to locate because of arbitrary orientation, alignment and illumination.

## 4. Conclusions

The problem of text location in images and video frames has been addressed in this paper. A method for text location based on multivalued image processing is proposed here. Applications of text location are presented which include text location in classified advertisement images, Web images, color images and video frames. Experimental results are given for different applications to show the performance of our algorithm. Compared to texture-based method [2] and motion-based approach for video [5], our method has a higher speed and accuracy in terms of the resulting bounding box of the text.

**Table 1. Text location Results.**

| Text carrier | #of test images | Typical size | Accuracy (%) | Avg. CPU time (s) |
|---|---|---|---|---|
| Advertisement | 26 | 548×769 | 99.2 | 0.15 |
| Web image | 54 | 385×234 | 97.6 | 0.11 |
| Color image | 30 | 769×537 | 72.0 | 0.40 |
| Video frame | 6,952 | 160×120 | 94.7 | 0.09 |

## References


[1] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 10:910–918, 1988.

[2] A. Jain and S. Bhattacharjee. Text segmentation using Gabor filters for automatic document processing. *Mach. Vision Applic.*, 5:169–184, 1992.

[3] A. Jain and B. Yu. Document representation and its application to page decomposition. *IEEE Trans. Pattern Anal. and Machine Intell.*, May 1998.

[4] A. K. Jain and B. Yu. Automatic text location in images and video frames. Technical Report MSUCPS:TR97-33, Dept. of Computer Science, Michigan State University, 1997.

[5] R. Lienhart and F. Stuber. Automatic text recognition in digital videos. In *Proc. of SPIE 2666*, pages 180–188, San Jose, 1996.

[6] I. Pitas and C. Kotropoulos. A texture-based approach to the segmentation of semitic image. *Pattern Recognition*, 25:929–945, 1992.

[7] S. N. Srihari, C. H. Wang, P. W. Palumbo, and J. J. Hull. Recognizing address blocks on mail pieces: specialized tools and problem-solving architectures. *Artificial Intelligence*, 8:25–35, 38–40, 1987.

[8] Y. Tang, S. Lee, and C. Suen. Automatic document processing: a survey. *Pattern Recognition*, 29:1931–1952, 1996.

[9] B. Yu and A. Jain. A generic system for form dropout. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18:1127–1134, 1996.

[10] B. Yu, A. Jain, and M. Mohiuddin. Address block location on complex mail pieces. In *Proc. of the 4th Int. Conf. on Document Analysis and Recognition*, pages 897–901, Ulm, 1997.

[11] Y. Zhong, K. Karu, and A. Jain. Locating text in complex color images. *Pattern Recognition*, 28:1523–1535, 1995.

[12] J. Zhou, D. Lopresti, and Z. Lei. OCR for World Wide Web images. In *Proc. of IS&T/SPIE Electronic Imaging: Document Recognition IV*, San Jose, 1997.

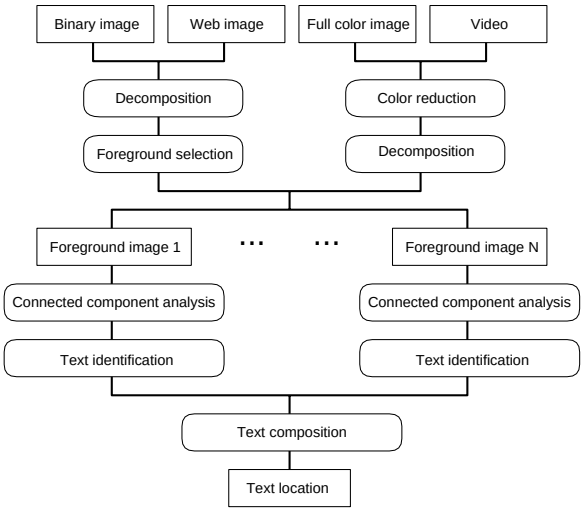


**Figure 1.** Automatic text location system.


**Figure 2.** Web images and located text regions.

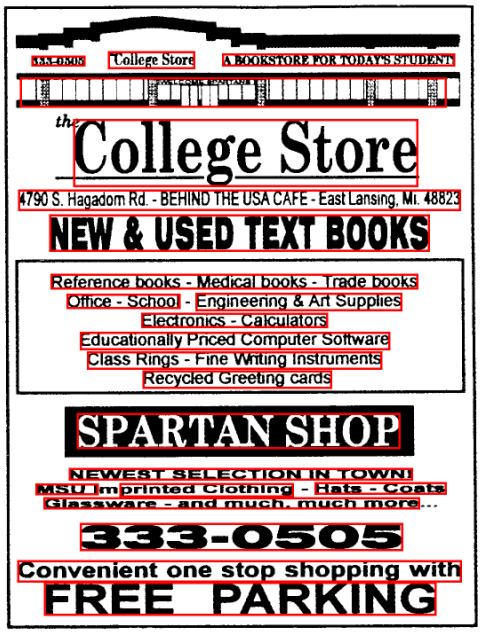


**Figure 3.** Located text for the advertisement images.


**Figure 4.** Text location in color images.


(a) (b)

(c) (d)

**Figure 5.** Video frames with low resolution.