Evaluation of Methods for Detection and Localization of Text in Video

S. Antani D. Crandall A. Narasimhamurthy V. Y. Mariano R. Kasturi
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{antani,crandall,kasturi}@cse.psu.edu

Abstract

The detection and recognition of text from unconstrained, general-purpose video is an important research problem with multiple applications in the surveillance, archiving and content-based retrieval contexts. Many text detection and localization algorithms have been proposed in the literature. However many of these algorithms either make simplistic assumptions as to the nature of the text to be found, or restrict themselves to a subclass of the wide variety of text that is observed in general purpose video. Almost all algorithms operate on images or individual video frames. It is also observed that the published results of most of these algorithms consist simply of sample images with bounded text boxes. There is a need for a quantitative evaluation of these algorithms against a challenging dataset. In this paper we present an evaluation of select text detection and localization algorithms. We present an evaluation of five algorithms. Some of these have been modified from the original work published by the authors. We discuss the method adopted for the evaluation and present results for the text localization methods. We observe that no one text detection and localization method is robust for detecting all kinds of text. It may be necessary to apply different methods that use independent heuristics to extract different kinds of text and then fuse these results temporally and across various algorithms.

1 Introduction

Indexing and efficient content-based retrieval of digital video has been identified as a challenging problem. Manual indexing and annotation, on the other hand, is a cumbersome task. Several automated methods have been developed which attempt to access image and video data by content from media databases [1]. A popular approach has been to temporally segment video into subsequences separated by shot changes, gradual transitions or special effects such as fade-ins and fade-outs [4]. In addition to these events and other objects contained within the scene imaged in the video, there is a considerable amount of text occurring in video which is a useful source of information. The presence of text in a scene, to some extent, naturally describes its content. If this text information can be harnessed, it can be used along with the temporal segmentation methods to provide a much truer form of content–based access to the video data. The current state of the art for extracting text from video either makes simplistic assumptions as to the nature of the text to be found, or restricts itself to a subclass of the wide variety of text that can occur in video of a general nature. Often, such methods only work on artificial text that is composited on the video frame. In addition, most video text extraction methods are simply methods for extracting text from images applied to single video frames and do not use the additional temporal information in video to good effect.

In the process of developing a system for extracting text from general purpose video, we realize that the different kinds of text have different heuristics. More than one algorithm is necessary to be able to detect all kinds of text appearing in the video. In order to improve the development process we studied the literature for text detection methods. It was seen that no formal evaluation of these methods had been done. The selected methods address different heuristics, thus improving chances for detecting the large variety of text seen in general purpose video. Some of these methods were then suitably enhanced or new methods created from the ideas contained in the original work.

In this paper we present a formal evaluation of these methods. We indicate the changes for the methods that have been modified. For the evaluation a ground truth has been developed. The paper is laid out as follows. We list some of the algorithms studied and those selected for evaluation in Section 2. Section 3 presents a discussion on issues involved with evaluation of text detection and localization algorithms. Section 4 describes the evaluation strategy and finally in Section 5 we present the results.

2 Previous Work

There is growing interest in the development of methods for detecting, localizing and segmenting text from video. We studied the methods published in the literature and applied some of the most promising methods to the system [5, 6, 7, 8, 10, 11, 13, 14, 16, 18, 19, 20]. The reader is referred to [17] for greater detail.

Of the methods seen in the literature, only those methods which we judged to be promising were selected. The selection was based on their applicability to general purpose video, use of features, ease of implementation and speed of detection. In addition to work done by others, we also include algorithms developed by us for evaluation. The algorithms chosen for evaluation are: **Method A** [3], **Method B** [9], **Method C**: based on initial idea published in [15], **Method D**: enhanced from initial idea published [2], and **Method E** [12]. Details on these methods can be found in the original work. We include details on the modifications here.

2.1 Modified Algorithm : Method C

Mitrea and de With [15] proposed a simple algorithm to classify video frame 4x4 pixel blocks into graphics or video based on the dynamic range and variation of gray levels within the block. We modified this method slightly and used it to classify blocks as text or non-text. The number of pixels in a 4x4 block that have similar gray levels is counted. If this number is less than a parameter and the dynamic range of the block is either less than or greater than two distinct thresholds, the block is classified as a text block.

2.2 Modified Algorithm : Method D

We have modified the method proposed by Chaddha et al [2] for classifying JPEG image blocks as text or non-text to work on MPEG-1 I-frames. While using only I-frames for detecting text is usually sufficient, for purposes of evaluation we operate the method on all I-, P- and B- frames. The DC coefficients are reconstructed for P- and B- frames. We have further refined it using an iterative thresholding scheme to improve performance. The method uses texture energy to classify 8x8 blocks as text or non- text and works as follows. An *a priori* subset of the 64 DCT coefficients in MPEG-encoded blocks is chosen. We chose the same subset used in the original paper. For each block, the sum of the absolute values of these coefficients is thresholded to categorize it as text or non-text.

The enhanced method builds on this idea as follows. A series of decreasing thresholds is iteratively applied from high to low and the appearance of more and more text blocks as the threshold is lowered is observed. Blocks that are classified as text at a particular threshold are kept if they also have a 8-neighbor that was classified as text at the previous higher threshold. The motivation for this is that text regions usually have at least one of their component blocks detected at the high threshold, so we can grow the text region by lowering the threshold without creating as many false positives. Any blocks with no neighbors on the left or right are removed. We also throw out any blocks which appear to be due to a sharp luminance change between two large homogeneous regions. This is done by averaging the DC term of the DCT coefficients three blocks to the left and right of a target block. The energy of these blocks is also averaged. If the average luminance of the three on the right is greater than that of the left by a certain threshold, or vice-versa, and the energies of the blocks are below a threshold, we conclude that this block was found because of a sharp luminance cliff and it is discarded. The final step is to apply the heuristic that text regions have to be wider than they are tall.

3 Evaluation of Text in Video : Issues and Discussion

Unlike the evaluation of automated methods for detection and localization of video events and objects contained within the imaged scene, the evaluation of text detection and localization methods presents interesting challenges. For example, when evaluating video shot change events [4], it is sufficient to detect at which frame a shot change (or other video transition event) occurred. The algorithms can be effectively and fairly evaluated on their performance. In case of localization of vehicles, faces or other objects a tightly fitting bounding region is typically effective enough for the application and a fair evaluation can be achieved.

In case of automated text detection and localization methods, however, the degree of correctness is difficult to determine. This is because the the intent of text detection and localization is to recognize it for indexing, retrieval and other purposes. Also, humans tend to identify the text contained in the video as characters and words along a line, sentences, and paragraphs. Unfortunately, the algorithms that detect "text-like" regions within the video frame do not take this approach into consideration when applying the heuristics. The algorithms detect small regions that contain text and the size of the region (tightness of fit) is dependent on the data element used by the algorithm. For example, algorithms that operate on MPEG DC coefficients, will result in regions along 8x8 block boundaries, while those that use horizontal windows of certain length will have different boundaries. The ground truth is usually marked by rectangular bounded regions which include the inter-character and sometimes inter-word non-text pixels. Also, non-text pixels surrounding the characters but within the ground-

truth bounded region are considered as text pixels. If an algorithm is very accurate and detects the text but not the surrounding or inter-character pixels, it suffers a penalty for being very precise in the form of a low recall (higher missed detections). Conversely an algorithm which operates on large blocks actually detects the text correctly but has a looser region boundary (due to operating block size) suffers the penalty in the form of low precision (higher false alarms).

Thus, in a sense, the algorithms are being evaluated unfairly. That is, they may be actually performing a little better that what is seen as a result of this evaluation. From our observations, we assess that the performance hit is approximately 5% in recall and precision.

4 Evaluation of Text Detection and Localization Algorithms

4.1 Test Data

Our test consists of nine MPEG-1 video sequences totaling 10299 frames. The sequences were captured at 30 frames per second and encoded in MPEG-1 with a 352x240 frame size. The sequences are portions of news broadcasts and commercials from various countries. The test database is challenging due to the poor quality and low contrast of these broadcasts. Text appears in a variety of colors, sizes, fonts, and language scripts.

The ground truth was performed frame-by-frame by humans using the ViPER tool from the University of Maryland. Bounding text box size, position, and orientation angle were specified to pixel-level accuracy. All regions distinguishable as text by humans were included in the ground truth, including text too small or fuzzy to be actually read but nevertheless identifiable as characters. Closely spaced words lying along the same horizontal were considered to belong to the same text instance. Separate lines of text were kept separate. The ground truth contains a total of 156 temporally-unique caption text instances (36491785 ground-truth pixels) and 146 scene text instances (57695829 ground-truth pixels). There are 302 text events in total.

4.2 Evaluation criteria

The ground truth defines tightly-bound text boxes for each frame. A good detection/localization algorithm would (ideally) produce similarly tight boxes. To evaluate the accuracy and tightness of fit of an algorithm's output, the pixel areas of the text regions in the ground truth are matched with the detected text regions. The evaluation is thus a frame-by-frame, pixel-by-pixel comparison of algorithm output with the ground truth. In case of non-horizontal oriented scene text, All pixels within the oriented bounding region are considered. During evaluation, each pixel in the test database is placed into one of three categories:

- **Detection:** Pixels belonging to text regions in the ground truth and regions identified as text by the localization algorithm.
- **False Alarm:** Pixels identified by the detection algorithm but not belonging to text regions in the ground truth.
- **Missed Detection:** Pixels belonging to the text regions in the ground truth and not identified by the algorithm.

The performance of an algorithm is quantified by its recall and precision, where:

$$Recall = \frac{detects}{detects + missed\,detects}$$
$$Precision = \frac{detects}{detects + false\,alarms}$$

Note that this pixel-level evaluation is very strict. Most actual applications would not require such precise localization. However our pixel-level criteria provides an easily measurable basis by which the relative performances of algorithms may be compared.

5 Results and Discussion

This section presents the results of the performance evaluation of the selected text detection and localization algorithms. Most of the parameters for the methods were kept as described in the original publication. Only those parameters which were highly dependent on the dataset were tuned on a small subset of the test dataset (approx. 1000 frames).

5.1 Performance Evaluation

Table 1 presents the caption text detection and localization performances, while Table 2 shows evaluation results for scene text, for the five algorithms on the entire test dataset. The table shows the raw numbers of total number of text pixels in the ground truth, the detected, false alarm, and missed detected pixels, along with computed recall and precision rates.

The results show that for caption text, overall Method D produces the highest precision rate of the individual algorithms, while the precisions of the other algorithms are comparably similar. Method E shows the highest recall. For scene text, Method D has the highest precision followed by

Algorithm	Text Pixels	Detects	FAs	MDs	Precision	Recall
Method A	36491785	14461593	62125359	22030192	39.63%	18.88%
Method B	36491785	14894707	45627542	21597078	40.82%	24.61%
Method C	36491785	15277954	59935070	21213831	41.87%	20.31%
Method D	36491785	26955906	119769022	9535879	73.87%	18.37%
Method E	36491785	17534331	35101135	18957454	48.05%	33.31%

Table 1. Detection/Localization Performance : Caption Text

Algorithm	Text Pixels	Detects	FAs	MDs	Precision	Recall
Method A	57695829	10016556	66570396	47679273	17.36%	13.08%
Method B	57695829	7283995	53238254	50411834	12.62%	12.04%
Method C	57695829	8398344	66814680	49297485	14.56%	11.17%
Method D	57695829	22207563	124517365	35488266	38.49%	15.14%
Method E	57695829	13878758	38756708	43817071	24.06%	26.37%

Table 2. Detection/Localization Performance : Scene Text

Method	Frames/sec.	Sec./frame
A	0.9	1.17
B	3.1	0.32
C	5.6	0.18
D	2.3	0.44
E	0.01	100

Table 3. Approximate algorithm running time.

Method E. Other methods have comparably similar results. Method E also has the highest recall for scene text.

The test database contains some very challenging scene text instances. For applications in surveillance and navigation, detecting scene text would be important. In other applications, such as video indexing, detecting scene text may not be important or even useful. Therefore scene text and caption text were evaluated separately. All of the algorithms perform better for caption text than the scene text.

The recall and precision rates of the algorithms in our evaluation are relatively low and perhaps highlight the need for better text detection and localization algorithms. Recently very high localization rates were presented in a method developed by Zhong et al [21]. We are presently evaluating that algorithm and propose the present the results at the workshop. A solution to improving the precision and recall values of the methods is to apply algorithm fusion to combine the outputs of multiple existing algorithms to produce better outputs. Each algorithm uses an independent set of features and heuristics and so a fusing of outputs of multiple algorithms is likely to be beneficial.

5.2 Running time

Table 3 gives approximate running times for our implementation of each of the algorithms on an SGI Octane workstation. The times include overhead resulting from I/O and MPEG stream decompression. The times are approximate since our implementations have not necessarily been fully optimized. Our implementation of the Method D was found to be the fastest (10.9 frames/sec.). This is in part because the MPEG stream need not be fully decompressed since it operates on raw DCT values. The value for this method in the table is 2.3 frames/sec. because P and B frames need to be decompressed completely and then DC coefficients need to be generated for these.

6 Conclusion

We have reviewed proposed text detection and localization algorithms. Several of the most promising were implemented and evaluated on a challenging dataset. The results show the strengths and weaknesses of the various algorithms. Methods D is the best algorithm in terms of precision and Method E the best in recall rate. We have also

highlighted the need for better detection and localization algorithms. Our current work includes investigating fusion algorithms which could intelligently integrate the outputs of several text detection algorithms, producing a better result than any individual algorithm.

References

[1] S. Antani, R. Kasturi, and R. Jain. Pattern Recognition Methods in Image and Video Databases: Past, Present and Future. In *Joint IAPR International Workshops SSPR and SPR*, number 1451 in Lecture Notes in Computer Science, pages 31–58, 1998.

[2] N. Chaddha, R. Sharma, A. Agrawal, and A. Gupta. Text Segmentation in Mixed–Mode Images. In *28th Asilomar Conference on Signals, Systems and Computers*, pages 1356–1361, October 1994.

[3] U. Gargi, S. Antani, and R. Kasturi. Indexing Text Events in Digital Video. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 916–918, August 1998.

[4] U. Gargi, R. Kasturi, and S. Antani. Performance characterization and comparison of video indexing algorithms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[5] H. Hase, T. Shinokawa, M. Yoneda, M. Sakaiand, and H. Maruyama. Character String Extraction by Multi–stage Relaxation. In *International Conference on Document Analysis and Recognition*, volume 1, pages 298–302, 1997.

[6] A. Hauptmann and M. Smith. Text, Speech, and Vision for Video Segmentation: The Informedia Project. In *AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision*, 1995.

[7] A. K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. *Pattern Recognition*, 31(12):2055–2076, 1998.

[8] H.-K. Kim. Efficient Automatic Text Location method and Content–Based Indexing and Structuring of Video Database. *Journal of Visual Communications and Image Representation*, 7(4):336–344, December 1996.

[9] F. LeBourgeois. Robust Multifont OCR System from Gray Level Images. In *International Conference on Document Analysis and Recognition*, volume 1, pages 1–5, 1997.

[10] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, 2000.

[11] R. Lienhart and F. Stuber. Automatic Text Recognition in Digital Videos. In *Proceedings of SPIE*, volume 2666, pages 180–188, 1996.

[12] V. Y. Mariano and R. Kasturi. Locating Uniform-Colored Text in Video Frames. In *Proc. International Conference on Pattern Recognition*, 2000.

[13] J. Ohya, A. Shio, and S. Akamatsu. Recognizing Characters in Scene Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:214–224, 1994.

[14] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith. Video OCR for Digital News Archive. In *IEEE International Workshop on Content–Based Access of Image and Video Databases CAIVD'98*, pages 52–60, January 1998.

[15] M. v. d. Schaar-Mitrea and P. H. N. de With. Compression of Mixed Video and Graphics Images for TV Systems. In *SPIE Visual Communications and Image Processing*, pages 213–221, 1998.

[16] J.-C. Shim, C. Dorai, and R. Bolle. Automatic Text Extraction from Video for Content-Based Annotation and Retrieval. In *Proc. International Conference on Pattern Recognition*, pages 618–620, 1998.

[17] U. Gargi and D. Crandall and S. Antani and R. Kasturi et al. A System for Automatic Text Detection in Video. In *International Conference on Document Analysis and Recognition*, 1999.

[18] L. Winger, M. Jernigan, and J. Robinson. Character Segmentation and Thresholding in Low-Contrast Scene Images. In *Proceedings of SPIE*, volume 2660, pages 286–296, 1996.

[19] V. Wu, R. Manmatha, and E. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(11):1224–1229, November 1999.

[20] Y. Zhong, K. Karu, and A. Jain. Locating Text in Complex Color Images. *Pattern Recognition*, 28(10):1523–1536, October 1995.

[21] Y. Zhong, H. Zhang, and A. K. Jain. Automatic Caption Localization in Compressed Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):385–392, 2000.