Improving Multimedia Retrieval with a Video OCR

Dipanjan Das[a] and Datong Chen[b] and Alexander G. Hauptmann[a]

[a] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA;
[b] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

We present a set of experiments with a video OCR system (VOCR) tailored for video information retrieval and establish its importance in multimedia search in general and for some specific queries in particular. The system, inspired by an existing work on text detection and recognition in images, has been developed using techniques involving detailed analysis of video frames producing candidate text regions. The text regions are then binarized and sent to a commercial OCR resulting in ASCII text, that is finally used to create search indexes. The system is evaluated using the TRECVID data. We compare the system's performance from an information retrieval perspective with another VOCR developed using multi-frame integration and empirically demonstrate that deep analysis on individual video frames result in better video retrieval. We also evaluate the effect of various textual sources on multimedia retrieval by combining the VOCR outputs with automatic speech recognition (ASR) transcripts. For general search queries, the VOCR system coupled with ASR sources outperforms the other system by a very large extent. For search queries that involve named entities, especially people names, the VOCR system even outperforms speech transcripts, demonstrating that source selection for particular query types is extremely essential.

Keywords: Video OCR, OCR, multimedia retrieval, video retrieval, Optical Character Recognition

1. INTRODUCTION

Analysis, indexing and retrieval of multimedia content requires a combination of various techniques. Other than speech recognition, image understanding, natural language processing and search strategies, video OCR (VOCR) systems have been observed to play a key role in multimedia information retrieval tasks.[1–3] Especially in broadcast news retrieval, appending overlay text to the index helps improve the search results by a great extent. The influence of the VOCR is more pronounced in case of queries involving named entities like names or locations. This is quite apparent because the text on a news screen usually denotes the salient entities that constitute the news item. One such example is shown in Figure 1 where a typical VOCR identifies a candidate text region by a series of image processing steps. After some post processing steps on the region, a binarized version of it (shown in top right) is passed on to an off-the-shelf OCR that identifies the text. This text is added to the index of transcripts used for retrieval purposes.

In the current work, we present a series of experiments with a video OCR system meant for a multimedia processing pipeline, and demonstrate that the system has improved search accuracy by a large extent. The novelty of the framework lies in the effective use of per image text extraction in multimedia retrieval. We compare the present system's retrieval performance with an earlier VOCR system,[4] that relies on multi-frame integration and combines evidence about a particular text region from consecutive or temporally proximate frames. Instead of multi-frame integration, the described system focuses on deeper analysis on individual frames, and is inspired by a work[5,6] that takes advantage of the structure of Roman script among other characteristics to efficiently extract candidate text regions from an individual image frame. We found that a multi-frame integration based approach would miss relevant overlay text when the text moves or flashes for a moment on the screen. To corroborate, our results exhibit that the VOCR system produces better accuracies for video retrieval – both from the perspective of recall and mean average precision, when the search index is built from the OCR alone. We include ASR transcripts to measure the effect of various sources over retrieval performance, and achieve a set of interesting results. Over and above general queries, we empirically show that the VOCR enhances retrieval accuracy for queries containing named entities and interestingly plays an even more significant role than speech transcripts, that form the primary component of the search index for video retrieval. All experiments have been performed on publicly available TRECVID 2005 and 2006 data.

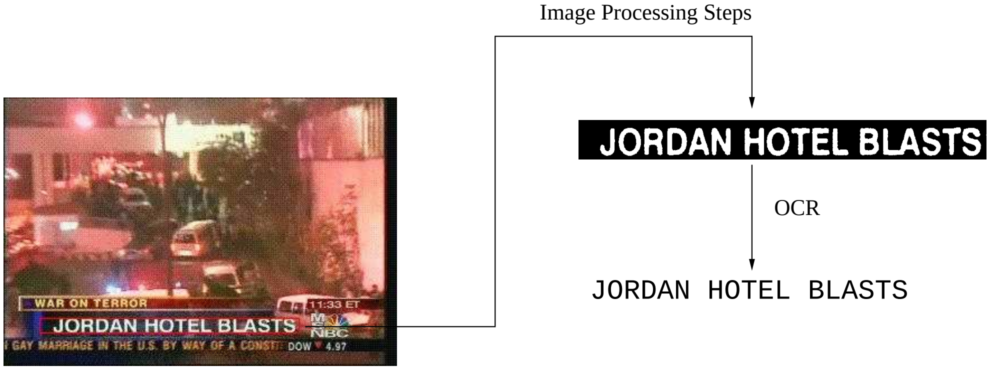


Figure 1. A screenshot from a broadcast news video. A typical image segmentation step extracts a prospective region of the image containing text. Running an OCR on the extracted region results in some text, that may be used for retrieval tasks.

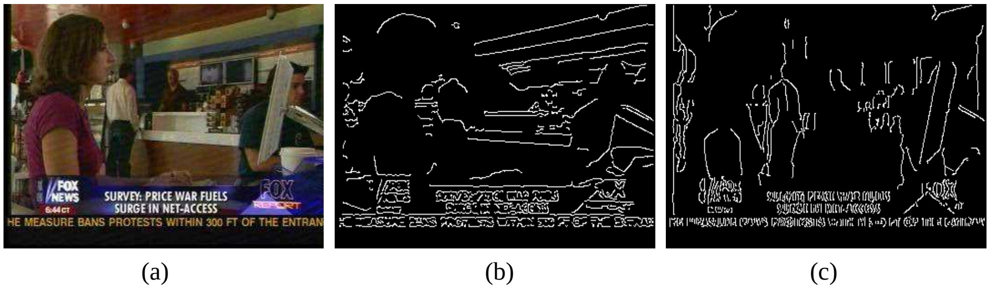
The approach taken in the present work can be categorized into a top-down approach of text recognition from images. There are primarily two broad phases of the entire task: a text detection phase and a text recognition phase. In the first phase, for a given image frame a cascade of various filters are applied to detect candidate text regions. A text localization step is adopted that extracts potential blocks of text on the image with considerable precision and high recall. To filter the text blocks from the candidates produced at this stage, various options can be considered. A machine learning approach is natural;[5] however, for reasons elaborated later in detail we rather choose to filter them using heuristics as much as possible. In the text recognition phase, candidate blocks of text are segmented to form a binary image, and fed into a commercial OCR to produce the final text.

The rest of the paper is organized as follows. Section 2 focuses on relevant previous work in the literature. Section 3 describes the VOCR framework, the algorithms for extracting text regions from individual frames and the techniques resulting in the actual text used for indexing the news video files. Section 4 enumerates the experiments and results achieved, followed by Section 5, where we conclude with a note about future work.

2. RELATED WORK

Efforts in the area of text extraction from images date back to research[7] involving localization of text on covers of journals and CDs. In this work it was assumed that text is concentrated in regions with high horizontal variance, possessing properties that can be exploited using connected component analysis. More relevant work[8] exist in the literature where techniques for skimming videos for significant segments were investigated using text detection techniques. On image frames, vertical edges were detected using a predefined template, and then they were grouped using a smoothing step. However, for both these methods if there is high horizontal contrast in the background, many false alarms may be produced. Texture segmentation based methods[9, 10] involve computation of texture features at each pixel from the derivatives of the image at different scales. In this work, pixels are classified into three classes in the feature space. The highest energy class denotes text, the other two indicate non-text and uncertainty. The segmentation method however is not robust to background noises and the texture feature computation is expensive. There have been research where candidate text regions are first located directly in the DCT compressed domain, and then they are reconstructed further refinement in the spatial domain.[11]Other related work[12] describes the detection of the top and bottom lines of the text occurring on the image, facilitating the localization of text.

In the domain of text based video retrieval, there has been a series of work where text extracted from the image frames constituting the video have been used for retrieval purposes. In a work[13, 14] on real video data, the authors have presented techniques of using approximate match and expansion of word semantics applied to the OCR outputs. However, the evaluation was not done on a standard dataset and other sources of the search index were not considered and compared. Multi-frame integration based methods for VOCR and applied to multimedia retrieval tasks[2] have been examined. We compare the developed VOCR with the one mentioned in this previous work, and we get significant improvements. Also, the work reports the use of every 3rd frame for



(a) (b) (c)

Figure 2. (a) shows an image frame from a video file. (b) denotes the image with horizontal edges (c) denotes the image with vertical edges.

analysis, while we use one frame per millisecond that makes the processing step much faster. Text localization and segmentation methods have been described in relevant work[15, 16] that make use of information integrated from consecutive frames; however again, extensive evaluation on standard data over a set of varied queries with different mix of sources have not been considered.

3. APPROACH

In this section, we describe the approach taken for extracting text from a video file. After sampling from a given video file at regular time intervals, each image frame is considered for text extraction. The method can broadly be divided into two major steps - text detection and text recognition. The following subsections describe these two steps in detail.

3.1 Text Detection

The step of text detection from an individual frame extracts candidate text regions using various image processing techniques. The algorithms that constitute this step are inspired by a previous work on extraction of overlay text from image frames.[5, 6] The entire task of detection of text can further be subdivided into two steps - *text localization* and *text verification*, which we describe in the rest of this subsection.

3.1.1 Text Localization

The text localization step extracts candidate text lines from a still image frame by applying a series of filters. The first step involves the identification of text blocks on the image, where the blocks are defined as rectangular regions containing one or more lines of text. Let S denote the set of pixels in an image I. For any site s ($s \in S$), we denote $T(s) = true$ if s is contained in a text block. Text blocks can be identified by estimating the probability $P(T(s) = true|I)$ and grouping together pixels with high probability into regions. There are two observations we can make about characters on images. Firstly, text blocks containing these characters have short edges in vertical and horizontal orientations. Secondly, the edges are connected to each other because of the connection between character strokes. These characteristics can be exploited using Canny filters that detect the vertical and horizontal edges.

$$C_v(s) = \begin{cases} 1 & \text{if } s \text{ is a vertical edge point} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$C_h(s) = \begin{cases} 1 & \text{if } s \text{ is a horizontal edge point} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Equations 1 and 2 denote the per site edge maps of an image. Figure 2 shows an image and the corresponding horizontal and vertical edge images after the Canny filter was applied to the original image. In the following step, different dilation operators are used for two kinds of edge pictures to connect the horizontal edges in the vertical direction and the vertical edges in the horizontal direction. The dilation operators are mathematical morphological operators. Their structures are shown in Figure 3 (a) and (b). These dimensions of the dilation

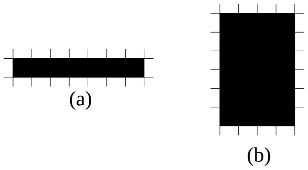


Figure 3. (a) 7×1 morphological operator for vertical edge dilation (b) 4×6 morphological operator for horizontal edge dilation.

operators are chosen empirically by observing the results of dilation using operators of various sizes. It is noticeable that the size of the vertical edge dilator is smaller in size than the horizontal operator because the horizontal edges lie closer to each other in comparison to vertical edges.

$$D_v = C_v \oplus rect_v \tag{3}$$

$$D_h = C_h \oplus rect_h \tag{4}$$

We denote the operators by $rect_v$ and $rect_h$ respectively and Equations 3 and 4 demonstrate their functions mathematically. Here, D_v and D_h are the dilated images respectively. The vertical and horizontal edge dilation results are shown in Figure 4 (a) and (b) respectively.

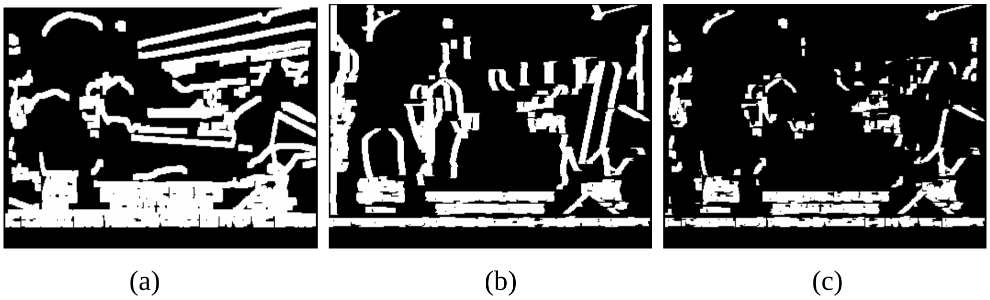


Figure 4. (a) Horizontal Edge Dilation.(b) Vertical Edge Dilation. (c) ANDed image.

Following the steps of vertical and horizontal edge dilation, there needs to be a method for creating a single image that captures the connectivity of vertical and horizontal edges. To this direction, we consider only the regions on the image frame that are covered by both the vertical dilated regions and the horizontal dilated regions. Mathematically this indicates taking a logical AND of the two dilated images. Therefore, $P(T|s, I)$ is estimated as:

$$P(T|, s, I) = D_v(s)D_h(s) \tag{5}$$

The resulting image is shown in Figure 4 (c). The entire procedure till this step is fast and extracts text regions irrespective of the intensity of the text and the backgrounds. Also, the parameters of Canny edge detection can be tuned by empirically observing the quality of text region detection over a small set of data. The text blocks are extracted from the image by doing a connected component analysis and computing the external contour of a region. A sample result is shown in Figure 5 (a). The recall at this step is quite high and the false positive regions are mainly slanting stripes and small areas of the background or human faces consisting of sharp edges. The next stage of filtering reduces their number by a large extent. To normalize size of the text suitable for a commercial OCR, the next step of text localization involves the identification of individual text lines. In other words, this subtask includes the identification of top and bottom baselines of horizontally aligned text lines. There are several advantages of this step. It reduces the number of false alarms such as slant stripes and also refines the location of text strings in candidate regions that contain text connected with background objects.

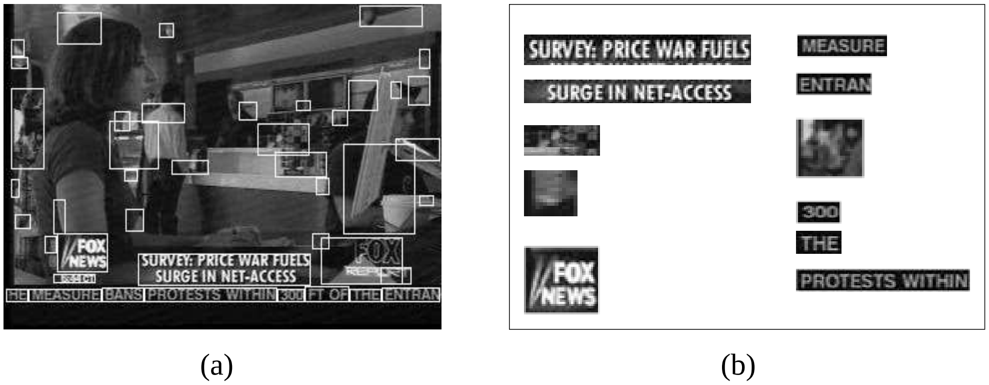


Figure 5. (a) Candidate text regions bordered by white rectangles. (b) Individual text lines extracted from the text regions.

The process of baseline detection starts with the computation of the Y-axis projection $h(y)$, where $h(y)$ denotes the number of text pixels in line y on the image. Three splitting algorithms are used in a cascading fashion after this that use the value of $h(y)$.

1. **Variable Length Splitting:** This method of text line identification aims at splitting a text region containing text strings of different lengths or text strings connected with background objects whose length is usually shorter than that of text strings. For a particular text region, the Y-coordinate y_0 with maximum absolute derivative is selected. If this maximum derivative is above a given threshold t_g and $h(y_0)$ is below 50% of the length of the longest line in the region, the region is split at line y_0.

2. **Equal Length Splitting:** When a region consists of two text lines of similar lengths, it may be split using Otsu's thresholding method.[17] Considering $h(y)$ as a one-dimension histogram, Otsu's method finds the threshold (line y_0) that minimizes the intra-class variance of the two text lines. Then, if $h(y_0)$ is less than 50% the longest line in this region, we split the region at line y_0.

3. **Baseline Refinement:** This method of splitting is applied when a text line cannot be split anymore. If a region cannot be split by the previous two algorithms, it is assumed that it may contain only one text line, and the top and bottom boundaries (baselines) of the region are refined to a yield more precise location. To this end, the greatest region (in height) is searched whose fill-factor is above a given threshold TF. Also, we threshold the width and height of a text line to prune the number candidate text regions.

Figure 5 (b) show final text lines extracted from an example image frame. The output of the text localization step may be optionally sent to a text verification step that we describe next.

3.1.2 Text Verification

The verification step of text detection involves the classification of extracted text lines into actual text regions or otherwise. Before running a classifier on an extracted line, each candidate text line is normalized with respect to height (16 pixels) to reduce the variance of character size. The features used for text verification fall into 4 categories - *Grayscale Spatial Derivative Features, Distance Map Features, Constant Gradient Variance Features* and *DCT Coefficient Features.* We describe them in some more detail as follows:

1. **Grayscale Spatial Derivative Features:** The first set of features are the spatial derivatives of the image brightness functions in both X and Y directions. It is important to note here that all features are extracted from 16 × 16 windows W sliding over the image grid S. Therefore, it has 512 dimensions. This feature measures the contribution of contrast for a given window.

2. **Distance Map Features:** Since grayscale values of both characters and background may vary, contrast is background dependent. To compensate for this weakness of the previous feature, the distance map (DM) features are used. DM is a feature image that only relies on the position of strong edges on the image. We define it as follows:
$$\forall(x,y) \in S, DM(x,y) = \min_{(x_i,y_i)\in E} d((x,y),(x_i,y_i)) \tag{6}$$
Here S is the image, (x,y) is an individual pixel, E is an edge and d is the Euclidean distance.

3. **Constant Gradient Variance Features:** Since distance maps are based on edges, they involve thresholds that need to be tuned. To avoid using any thresholds, constant gradient variance (CGV) features are used. This variance normalization technique is usually used to enhance grayscale images or to preprocess input data in speech. The basic underlying principle is to apply this technique on the gradient image to normalize the contrast at a given point using the local contrast variance computed in a neighborhood of this point. More formally, let $g(x,y)$ be the gradient magnitude at pixel (x,y) and let $LM(x,y)$ (or $LV(x,y)$) denote the local mean (or local variance) of the gradient. We define them as:
$$LM(x,y) = \frac{1}{|G_{x,y}|} \sum_{(x_i,y_i)\in G_{x,y}} g(x_i,y_i) \tag{7}$$
$$LV(x,y) = \frac{1}{|G_{x,y}|} \sum_{(x_i,y_i)\in G_{x,y}} (g(x_i,y_i) - LM(x,y))^2 \tag{8}$$
where $G_{x,y}$ is defined as a 9×9 area around (x,y). Finally, we define the CGV for a site (x,y) is as:
$$CGV(x,y) = (g(x,y) - LM(x,y))\sqrt{\frac{GV}{LV(x,y)}} \tag{9}$$
where GV denotes the global gradient variance defined over the whole image S.

4. **DCT Coefficients Features:** DCT coefficients are computed over 16×16 blocks using fast DCT algorithms. This frequency domain feature is widely used in JPEG and MPEG compression techniques.

Since, the features are calculated for a sliding window of 16×16 size, there are a total of 1280 features per feature vector when all types of features are concatenated. SVMs are used for classification[18] and Gaussian Radial Basis Functions are used for experimentation. The standard library LibSVM[19] is used as a tool for classification. A total of 10,000 feature vectors are used for training. For a particular region $\mathbf{r}$, we define the confidence value of whether it is a text region as:
$$Conf(\mathbf{r}) = \sum_{\mathbf{z}_i^r \in Z_r} S(\mathbf{z}_i^r).\frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{-d_i^2}{2\sigma_0^2}} \tag{10}$$
where $\mathbf{z}_i^r$ is a feature vector for a window on the region $\mathbf{r}$, and $S(\mathbf{z}_i^r)$ is the sign of the classification on the feature vector by the SVM classifier. d_i is the distance from the geometric center of the ith sliding window to the geometric center of the text line $\mathbf{r}$, and σ_0 is a scale factor depending on the text line length. If $Conf(\mathbf{r}) \geq 0$, we classify the text region as a valid region, otherwise we reject it.

3.1.3 Study on the Text Verification Step

We performed a short study on the performance of the entire text detection procedure, estimating its time efficiency. Since the text verification step involves using SVMs for a very large space of 1280 features per image window W, the classification model file becomes very large ($\sim$ 80MB for a training set size of 10,000). The classification time also is pretty high between 2 and 3 seconds per video frame. Since the text verification step only helps improving the precision of a set of candidate text regions keeping the region extraction recall as high as possible, we decided to study the effect of removing the verification step. In other words, this just means using

Metric	VOCR with Text Verification	VOCR without Text Verification
Time taken to convert the video to text regions	1352 seconds	41 seconds
Time taken to convert text regions to final text	11 seconds	60 seconds
Total time taken to convert from video to final text	1363 seconds	101 seconds
Text files produced	588	1648

Table 1. Performance of the VOCR with and without the text verification step

the output of the text localization step, binarizing each text line and sending them to the commercial OCR. This makes sense because for candidate text regions containing no text, the OCR normally outputs nothing. However, if the OCR does produce some garbage output for an invalid text region, the text will have negligible probability in influencing retrieval results. This is because of the fact that the garbage output will never match a keyword in a search query, thus keeping the search results the same. After the removal of the verification step, it was observed that the total time taken to produce the final OCR transcripts for a given video file was many times less than the total time taken to produce the transcripts with the verification step. This implies that:

$$(\textit{time taken to remove invalid text regions using the verification step} + \\ \textit{time taken to OCR the resulting text regions}) \gg \\ (\textit{time taken to OCR all candidate text regions including invalid frames})$$

Table 1 shows the results of a small experiment performed on a part of a broadcast news video. The experiment was performed only on a sample of 3871 frames, among which only one frame per millisecond was considered. The results clearly show that the process is faster when text verification is not performed, and we chose that alternative in the current work.

This is a situation unique to a video OCR system coupled with a retrieval pipeline, where a low precision process is not penalized. Recall is the only metric that plays an important role. Because of this reason, we decided to leave text verification out of the pipeline and feed the next step of text recognition with the binarized output of text localization directly. However, we conjecture at this point that tuning SVM parameters and performing more feature engineering might revert the above relationship, but we did not perform that study in the current work.

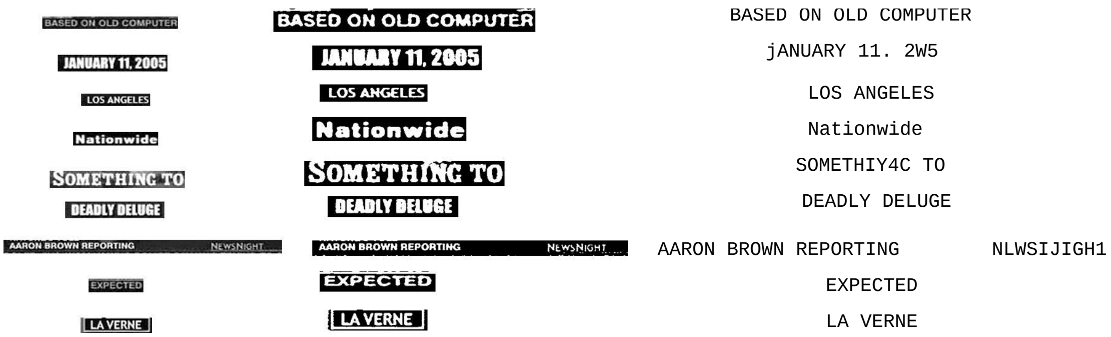
3.2 Text Recognition

Compared to the lengthy process of text localization, text recognition is straightforward. Before passing on a text line to a commercial OCR, a significantly important step of binarization of the image has to be performed to convert the image to an OCR readable format. Instead of going to complex segmentation methods, we simply use Otsu's algorithm[17] for binarization of the image. The underlying principle of Otsu's algorithm is to create a histogram of the image and select a threshold to maximize interclass variance. This produces a binary image, suitable for input to a commercial OCR.[20] Figure 6 shows a set of text line regions extracted from real video frames, the corresponding binarized images and the text extracted from the commercial OCR. It is noticeable from the figure that at times, the output from the commercial OCR is incorrect resulting in information loss.

The text produced as output from the commercial OCR is used as the VOCR transcript. The following section describes the experiments done with the Video OCR thus produced, with other transcription data and our video dataset.

4. EXPERIMENTS AND RESULTS

In this section, we first describe the data used to evaluate the performance of the new VOCR. Second, we explain the framework of the experiments, followed by the performance of the module compared to other methods of transcribing news video.



```
BASED ON OLD COMPUTER
jANUARY 11. 2W5
LOS ANGELES
Nationwide
SOMETHIY4C TO
DEADLY DELUGE
AARON BROWN REPORTING        NLWSIJIGH1
EXPECTED
LA VERNE
```

Figure 6. Extracted text lines, corresponding binarized images and text extracted from a commercial OCR

4.1 Data

We evaluate the video OCR on an assortment of TRECVID 2005 and 2006 evaluation datasets*. In the present set of experiments, we have tested our approach on the English language video files. The videos are from English news channels and have significant amount of overlay text characterizing broadcast multimedia data. Altogether the data set consists of 147 video files, corresponding to 30 hours and 52 minutes of video from TREC'05 data and 52 hours and 37 minutes of video from TREC'06. These video files are processed by the VOCR module to result in a collection of transcripts, each for one millisecond of a video file. We accumulate these transcripts on the basis of shots to create shot documents.

Along with the VOCR transcripts, we have generated other textual transcripts from Sato's VOCR system[4] and an automatic speech recognition system (ASR).[21] The ASR system is a standard one developed by Microsoft, allows speech recognition and speech synthesis within Windows applications, and can be used by third party applications. For each video file, our pipeline uses the ASR system to produce words at intervals at millisecond precision. Thus, for every video file, we have a list of words with timestamps attached. Using a database of shot start and end times, we create one transcript each for a shot, similar to the fashion in which we create shot documents for the video OCR.

Sato's VOCR uses analysis of consecutive frames to extract overlay text on the screen. In contrast, we do not accumulate information over multiple frames but rather perform deeper analysis on independent individual frames. We conjecture that analysis of single frames would result in better retrieval because flashing or moving overlay text would not be captured by Sato's VOCR system. This is corroborated by the results that we describe in the following subsections. Datasets corresponding to different combinations of these sources are considered for evaluation establishing a comparison of their influence on shot retrieval.

Metric	$VOCR_{MA}$	$VOCR_{FD}$
Time taken to the process the file	3767 seconds	1191 seconds
Text regions produced	78	4373
Text regions resulting in OCR text	40	2305
Per frame text region recall	-	$\sim 99\%$

Table 2. Performance of $VOCR_{MA}$ and $VOCR_{FD}$ on various metrics for a video file of 5 minutes 37 seconds

4.2 Experimental Setup

As part of our experimental setup, we created the following combination of sources for a given shot: 1) Sato's VOCR with text accumulation over multiple frames ($VOCR_{MA}$ henceforth) only, 2) the newly implemented VOCR with per frame deep analysis ($VOCR_{FD}$ henceforth) only, 3) $VOCR_{MA}$ and $VOCR_{FD}$, 4) ASR only,

*http://www-nlpir.nist.gov/projects/trecvid/

Source	**MAP**		**Recall**	
	TREC05	TREC06	TREC05	TREC06
$VOCR_{MA}$	0.0003	0.0053	0.0033 (*baseline*)	0.0163 (*baseline*)
$VOCR_{FD}$	0.0015	0.0091	0.0170 (+**415.2%**)	0.0277 (+**69.9%**)
$VOCR_{FD} + VOCR_{MA}$	0.0014	0.0098	0.0174 (+**427.3%**)	0.0327 (+**100.6%**)

(a)

Source	**MAP**		**Recall**	
	TREC05	TREC06	TREC05	TREC06
ASR	0.0111	0.0045	0.0353(*baseline*)	0.0413(*baseline*)
$ASR + VOCR_{MA}$	0.0112	0.0071	0.0379(+7.3%)	0.0520(+25.9%)
$ASR + VOCR_{FD}$	0.0045	0.0099	0.0466(+**32.0%**)	0.0570(+**38.0%**)
$ASR + VOCR_{FD} + VOCR_{MA}$	0.0044	0.0105	0.0470(+**33.1%**)	0.0586(+**41.9%**)

(b)

Table 3. (a) MAP and Recall values for different sources (b) MAP and Recall values for different sources with ASR transcripts. Improvements over baselines are shown in parenthesis.

5) ASR and $VOCR_{MA}$, 3) ASR and $VOCR_{FD}$, and 5) ASR, $VOCR_{MA}$ and $VOCR_{FD}$. We index the shots from each of these combinations using the Lemur Toolkit,[22] after removing stop words from the transcript texts.

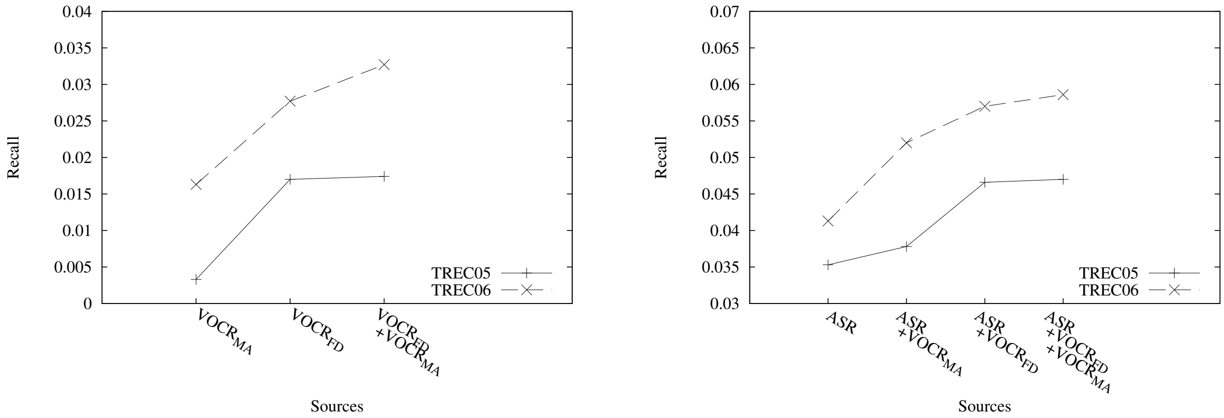
To compare the performances of $VOCR_{FD}$ and $VOCR_{MA}$, we performed both VOCRs on an broadcast news MPEG clip of length 5 minutes 37 seconds. $VOCR_{FD}$ took 1191 seconds while $VOCR_{MA}$ took 3767 to analyze the clip. $VOCR_{MA}$ produces 78 text regions from the clip, among which 40 produce some text from the commercial OCR. On the other hand, $VOCR_{FD}$ produces 4373 text regions, among which 2305 produce some text from the commercial OCR. The result shows that the $VOCR_{FD}$ has a superior performance and that is corroborated by the results of multimedia retrieval that we describe in the following subsection. We also observed that $\sim 50\%$ of the text regions were false positives for $VOCR_{FD}$, but it had a $\sim 99\%$ recall, which is what matters for retrieval purposes. Table 2 enumerates these details.

To complete our setup on evaluating the eventual video retrieval performance, we possess a set of 48 queries, constituted by 24 queries each from TRECVID 2005 and 2006 datasets respectively. We performed a simple query processing by removing all stop words from them and ran the queries on the Lemur created index.

4.3 Results

We have Mean Average Precision (MAP) and Recall at 1000 results as our primary measures of comparison between the various transcript sources. Table 3 shows the trends in MAP measures for the two datasets. It is clear from the recall values in the table that $VOCR_{FD}$ alone performs much better than $VOCR_{MA}$ alone with an improvement of 415% and 70% for the TREC'05 and TREC'06 datasets respectively. Also, the combination of $VOCR_{FD}$ and ASR performs much better (35% average improvement over the ASR baseline) than a mix of $VOCR_{MA}$ and ASR (16.6% average improvement over the ASR baseline). Finally, from the recall perspective we observe that the combination of all three sources results in the best measure. Figure 7 plots this graphically.

We expect the MAP measure to follow similar trends; if we just look at the figures for $VOCR_{FD}$ and $VOCR_{MA}$ alone, we observe considerable improvements for the former case. However, for a mix of ASR and the OCRs give contradictory results for the two datasets. We investigated this by looking at the query types for the two datasets. We observed that for a particular query in the TREC05 dataset ("Find shots of Tony Blair"), the ASR system has an average precision of 0.22 which is about 20 times the MAP of the ASR index for TREC05. The corresponding average precision for the OCRs is zero. This skews the results for the entire TREC05 dataset. However, if we look at the TREC06 dataset where results were not affected by a sole query,



(a) (b)

Figure 7. (a) Recall values for different sources (b) Recall values for different sources with ASR transcripts

we get a clear picture that $VOCR_{FD}$ performs better than $VOCR_{MA}$; a mix of the three sources performs the best as the recall case.

It would be wise to note that the observed MAP values are relatively low for all the experiments. The values are such because we have considered only individual shots and not smoothed scores over several temporally proximate shots, unlike a full video retrieval engine. Moreover, in a real video retrieval system,[23] color features are taken into consideration, semantic classification of scenes and query type analysis are performed to achieve much higher MAP scores. Since the focus of our work was to only evaluate the performance of the developed VOCR in comparison to a similar existing system, we did not test the framework in a full-featured retrieval scenario.

We also evaluated the results on queries that contain named entities (NEs). We conjecture that $VOCR_{FD}$ would perform better on average for such queries than all other individual sources and it was corroborated by the recall values in particular. The MAP values for TREC06 suggest a similar trend as well. However, again because of the influence of the aforementioned single query the MAP values of TREC05 gets skewed towards the ASR transcripts. Table 4 shows the different figures for MAP and Recall for NE type queries.

5. CONCLUSION AND FUTURE WORK

In this paper, we have put forward a VOCR system inspired by a previous detailed work on text extraction from images. We focused on the aspects of the text extraction process that are relevant for video information retrieval. A comparison with another VOCR implemented using multi-frame integration and other types of image processing techniques has also been done, which empirically shows the superiority of the VOCR system from the perspective of multimedia retrieval. The effectiveness of the VOCR for particular queries is also observable from the enumerated results. The VOCR system has been integrated to the Informedia Digital Library Project,[23] was a part the TREC 2006 Video Track evaluations and is being used in a daily basis.

However, we have not performed enough feature engineering for the text verification step, inclusion of which leads to slow extraction of text regions. There is a lot of scope of work in that direction. The process of text recognition can be improved by a large extent by using more detailed techniques of image segmentation and the separation of text from complex backgrounds. From the information retrieval side, we have not performed enough query processing and weighting of different sources for different types of queries. Appending these features to the VOCR might improve the retrieval results to even greater extents. Other retrieval techniques, like using different weights for different sources, processing of VOCR results by using a dictionary and nearest word match might improve search results further.

Source	**MAP**		**Recall**	
	TREC05	TREC06	TREC05	TREC06
$VOCR_{MA}$	0.0009	0.030	0.026($baseline$)	0.084($baseline$)
$VOCR_{FD}$	0.0027	0.047	0.104(+**300.0%**)	0.128(+**52.4%**)
$VOCR_{FD} + VOCR_{MA}$	0.0028	0.055	0.111(+**326.9%**)	0.156(+**85.7%**)

(a)

Source	**MAP**		**Recall**	
	TREC05	TREC06	TREC05	TREC06
ASR	0.0292	0.023	0.052($baseline$)	0.112($baseline$)
$ASR + VOCR_{MA}$	0.0295	0.037	0.072(+38.5%)	0.168(+50.0%)
$ASR + VOCR_{FD}$	0.0094	0.052	0.118(+**126.9%**)	0.186(+**66.1%**)
$ASR + VOCR_{FD} + VOCR_{MA}$	0.0095	0.056	0.124(+**138.5%**)	0.191(+**70.1%**)

(b)

Table 4. (a) MAP and Recall values for different sources for NE type queries (b) MAP and Recall values for different sources for NE type queries with ASR transcripts. Improvements over baselines are shown in parenthesis.

REFERENCES

1. D. Chen, H. Bourlard, and J.-P. Thiran, "Text identification in complex background using svm.," in *CVPR (2)*, pp. 621–627, 2001.
2. A. G. Hauptmann, R. Jin, and T. D. Ng, "Multi-modal information retrieval from broadcast video using ocr and speech recognition," in *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pp. 160–161, ACM Press, (New York, NY, USA), 2002.
3. A. Hauptmann, R. Jin, and T. Ng, "Video retrieval using speech and image information," in *Electronic Imaging Conference (EI'03), Storage and Retrieval for Multimedia Databases*, 2003.
4. T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in *CAIVD*, pp. 52–60, 1998.
5. D. Chen, *Text detection and recognition in images and video sequences.* PhD thesis, École Polytechnique Fédérale de Lausanne, Aug. 2003.
6. D. Chen, J.-M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recognition* **37(3)**, 2004.
7. Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," in *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, p. 146, IEEE Computer Society, (Washington, DC, USA), 1995.
8. M. Smith and T. Kanade, "Video skimming for quick browsing based on audio and image characterization," Tech. Rep. CMU-CS-95-186, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, July 1995.
9. V. Wu, R. Manmatha, and E. M. Riseman, "Finding text in images," in *ACM DL*, pp. 3–12, 1997.
10. V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: An automatic system to detect and recognize text in images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(11), pp. 1224–1229, 1999.
11. Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(4), pp. 385–392, 2000.
12. K. Sobottka, H. Bunke, and H. Kronenberg, "Identification of text on colored book and journal covers," in *Proceedings of the 5th International Conference on Document Analysis and Recognition*, pp. 57–62, 1999.
13. H. Li and D. S. Doermann, "Video indexing and retrieval based on recognized text.," in *IEEE Workshop on Multimedia Signal Processing*, pp. 245–248, 2002.

14. H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital videos," *IEEE Transactions on Image Processing* **9(1)**, pp. 147–156, 2000.
15. R. Lienhart and W. Effelsberg, "Automatic text segmentation and text recognition for video indexing," *Multimedia Syst.* **8**(1), pp. 69–81, 2000.
16. R. Lienhart and A. Wernicked, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits and Systems for Video Technology* **12**, pp. 236–268, 2002.
17. N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Systems, Man and Cybernetics* **9**, pp. 62–66, 1979.
18. R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of svms for very large scale problems," in *Advances in Neural Information Processing Systems*, MIT Press, 2002.
19. C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.
20. "Textbridge ocr." http://www.nuance.com/textbridge/.
21. "Microsoft speech API." http://www.microsoft.com/speech/default.mspx.
22. "Lemur toolkit for language modeling and information retrieval." http://www.lemurproject.org/.
23. "The Informedia digital video library." http://www.informedia.cs.cmu.edu.