To appear in the *Fourth ACM International Multimedia Conference*, Boston, USA, November 1996

REIHE INFORMATIK
6/96

Indexing and Retrieval of Digital Video Sequences based on Automatic Text Recognition

Rainer Lienhart
Universität Mannheim
Praktische Informatik IV
L15,16
D-68131 Mannheim

revised November 1996

Automatic Text Recognition for Video Indexing

Rainer Lienhart

University of Mannheim
Praktische Informatik IV
68131 Mannheim, Germany
lienhart@pi4.informatik.uni-mannheim.de
phone: +49-621-292-3381 fax: +49-621-292-5745

ABSTRACT

Efficient indexing and retrieval of digital video is an important aspect of video databases. One powerful index for retrieval is the text appearing in them. It enables content-based browsing. We present our methods for automatic segmentation and recognition of text in digital videos. The algorithms we propose make use of typical characteristics of text in videos in order to enable and enhance segmentation and recognition performance. Especially the inter-frame dependencies of the characters provide new possibilities for their refinement. Then, a straightforward indexing and retrieval scheme is introduced. It is used in the experiments to demonstrate that the proposed text segmentation and text recognition algorithms are suitable for indexing and retrieval of relevant video scenes in and from a video database. Our experimental results are very encouraging and suggest that these algorithms can be used in video retrieval applications as well as to recognize higher semantics in video.

KEYWORDS: video processing, character segmentation, character recognition, OCR, video indexing, video content analysis

1 INTRODUCTION

There is no doubt that video is an increasingly important modern information medium. Setting free its complete potential and usefulness requires efficient content-based indexing and access. One powerful high-level index for retrieval is the text contained in videos. This index can be built by detecting, extracting and recognizing such text. The index enables the user to submit sophisticated queries such as a listing of all movies featuring John Wayne or produced by Steven Spielberg. Or it can be used to jump to news stories about a specific topic, since captions in newscasts often provide a condensation of the underlying news story. For example, one can search for the term “Financial News” to get the financial news of the day. The index can also be used to record the broadcast time and date of commercials, helping the people who check for their clients whether their commercial has been broadcasted at the arranged time on the arranged television channel. Many other useful high-level applications are imaginable if text can be recognized automatically and reliably in digital video.

In this paper we present our methods for automatic text segmentation and text recognition in digital videos. We also demonstrate their suitability for indexing and retrieval. For better segmentation and recognition performance our algorithms analyse typical characteristics of text in video. Inter-frame dependencies of text incidences promise further refinement. Text features are presented in Section 2, followed by a description of our segmentation and recognition algorithms in Section 3 which are based on the features stated in Section 2. Then, in Section 4 we introduce a straightforward indexing and retrieval scheme, which is used in our experiments to demonstrate the suitability of our automatic text recognition algorithms for indexing and retrieval. The experimental results of each step - segmentation, recognition and retrieval - are discussed in Section 5. They are investigated independently for three different film genres: feature films, commercials and newscasts. Section 6 reviews related work, and Section 7 concludes the paper.

2 TEXT FEATURES

Text may appear anywhere in the video and in different contexts. It is sometimes a carrier of important information, at other times its content is of minor importance and its appearance is only accidental. Its significance is related to its nature of appearance. We discriminate between two kinds: *scene text* and *artificial text.* Scene text appears as a part of and was recorded with the scene, whereas artificial text was produced separately from the video shooting and is overlaid over the scene in a post-processing stage, e.g. by video title machines.

Scene text (e.g. street names or shop names in the scene) mostly appears accidentally and is seldom intended. How-

ever, when it appears unplanned it is of minor importance and generally not suitable for indexing and retrieval. Moreover, due to its incidental and the thus resulting unlimited variety of its appearance, it is hard to detect, extract and recognize. It seems to be impossible to identify common features, since the characters can appear under any slant, tilt, in any lighting and upon straight or wavy surfaces (e.g. on a T-shirt). It may also be partially occluded.

In contrast, the appearance of artificial text is carefully directed. It is often an important carrier of information and herewith suitable for indexing and retrieval. For instance, embedded captions in TV programs represent a highly condensed form of key information on the content of the video [18]: there, as in commercials, the product and company name are often part of the text shown. (Here, the product name is often scene text but used like artificial text!) Therefore, in this paper we concentrate on extraction of artificial text. Fortunately, its appearance is subjected to many more constraints than that of scene text since it is made to be read easily by viewers.

The mainstream of artificial text appearances is characterized by the following features:

- Characters are in the foreground. They are never partially occluded.
- Characters are monochrome.
- Characters are rigid. They do not change their shape, size or orientation from frame to frame.
- Characters have size restrictions. A letter is not as large as the whole screen, nor are letters smaller than a certain number of pixels as they would otherwise be illegible to viewers.
- Character are mostly upright.
- Characters are either stationary or linearly moving. Moving characters also have a dominant translation direction: horizontally from right to left or vertically from bottom to top.
- Characters contrast with their background since artificial text is designed to be read easily.
- The same characters appear in multiple consecutive frames.
- Characters appear in clusters at a limited distance aligned to a horizontal line, since that is the natural method of writing down words and word groups. But this is not a prerequisite, just a strong indicator. From time to time just a lone character might appear on one line.

Our text recognition algorithms are based on these features. However, they also take into account that some of these features are relaxed in practice due to artifacts caused by the narrow bandwidth of the TV signal or other technical imperfections.

3 TEXT RECOGNITION

Our feature-based text recognition approach is performed in two steps: text segmentation and text recognition.

The first step, text segmentation, extracts all pixels out of the video that are part of text characters and discards all pixels which do not belong to characters. In practice, since we do not know so far, which pixels belong to characters and which do not, the first step discards only those pixels which are most probably not part of text characters. Thus, as a result we get candidate character regions which may be characters or parts of characters.

The second step, text recognition, then tries to recognize the characters contained in the candidate character regions by applying optical character recognition techniques.

3.1 Text Segmentation

The text segmentation algorithms rely completely on the character features stated in the previous section. One important feature is contrast: designed for and perceived by human viewers. To adjust contrast and color difference calculation to human perception, we transform all frames into the nonlinear R'G'B' color system [12] before applying any segmentation algorithm. Although nonlinear R'G'B' is not optimized for perceptual uniformity like CIE L*a*b* [12], it is a good compromise between computational burden and perceptual uniformity requirement. The demand for operating on color frames and converting them into a more perceptually uniform color system is one deep insight we gained from our experiments, and constitutes one major difference from the segmentation algorithms described in [6].

The segmentation is performed as follows:
In a first preprocessing step the number of different colors used in each video frame is reduced. This transformation does not affect the outline of the characters since characters are assumed to be monochrome and contrasting with their background. However, it generates larger homogeneous regions, thereby reducing the complexity of each frame, easing subsequent processing
We employ the Split-and-Merge algorithm proposed by Horowitz and Pavlidis to perform segmentation [5]. It is based on a hierarchical decomposition of a frame. The split process begins with the entire image as the initial segment, which is then split into quarters. Each quarter is tested against a certain homogeneity criterion to determine whether the segment is "homogeneous enough". If not homogeneous enough, the segment is split again into quarters. This process is applied recursively until only homogeneous segments are left. We use a threshold of the Euclidean distance of the color values as homogeneity criterion. A homogeneous segment is assigned its average color. Next, in the merge process, adjacent segments are merged together if their average color difference is less than a threshold (Figure 1 (c)). This algorithm is known to produce good segmentation results in most cases.

The following steps reduce the number of candidate character regions. Some regions are too large and others are too small to be instances of characters. Therefore, (monochrome) regions whose width and height exceed a threshold *max_size* are removed, as are connected monochrome regions whose combined expansion is less than a threshold *min_size* (Figure 1 (d)).

Since we are analyzing text in videos that has been generated by video title machines, the same text typically appears in a number of consecutive frames. Thus, by means of motion analysis we should be able to find the same text characters in consecutive frames. Therefore, for each region in a frame, we search for one in the consecutive frame that corresponds in size, color and shape. If we are unable to find a corresponding one in the next frame, the region is regarded as a non-character segment and discarded. Fortunately, char-

(a) Frame n

(b) Frame n+1

(c) Applying split & merge

(d) Applying size restriction

(e) Applying motion analysis

(f) Applying contrast analysis

Figure 1: Text segmentation: The different processing steps.

acters are rigid. Thus, simple region-matching can be applied to find corresponding characters and words. Since characters - as already mentioned - are either stationary or moving linearly, this condition is loosely checked over five consecutive frames (Figure 1 (e)).

Finally, each remaining candidate character region is checked for contrast with its surroundings. If no such contrast, even only a partial one, is found, we conclude that the region cannot belong to a character. Consequently, the region is discarded (Figure 1 (f)).

In contrast to our previous proposal in [6] we do not apply any width-to-height ratio constraints to clustered regions. It has turned out that width-to-height ratio constraints are effective in manual tune-ups, but it is impossible to find suitable values coping with all artificial text appearances. The thresholds are either too restrictive, thus deleting character regions of some text appearances, or are too loose to have any effect.

3.2 Text Recognition

The segmentation step delivers a video showing candidate character regions. In principle, any standard OCR software can now be used to recognize the text in the segmented frames. However, a hard look at the properties of the candidate character regions in the segmented frames reveals that most OCR software packages will have significant difficulty to recognize the text.

For each segmented frame the following properties are true:

- characters may consist of several regions with different color values.
- several regions may be connected to each other, and these regions may belong to several different characters and/or non-character regions.
- characters are not cut out precisely, even though it seems so to the human eye.

These properties pose major difficulties for standard OCR software, which is optimized for recognizing text from scanned print media where the background of text is intentionally kept simple. The main problems are: Firstly, many OCR systems assume text to be printed in black on a white background. Therefore, they demand black-and-white images as input. However, our segmented frames are color images, and it is not straightforward how to convert them into black-and-white bitmaps, especially since both characters and background can be of any color. Thus, an OCR system operating on gray-scale images is the minimum requirement. Unfortunately, these systems also assume a monochrome text on a monochrome background, so that they just extract characters regions by thresholding the gray-scale image. Again, this is not appropriate as we can see in Figure 2. Secondly, segmented characters are not cut out exactly. Often they are surrounded by some kind of "aura" belonging partially to the background and partially to a character. These "aura" pixels cannot be related to background or character by simple thresholding or region-growing. They can only be classified as background or character if an iterative character classification approach is used. Thus, this "aura" also represents a significant problem for most of the currently available OCR software. Their limited suitability for our purpose stems from the fact that they expect an input with a different feature set.

Figure 2: Two enlargements of segmented characters.

Therefore, we have implemented our own OCR software, allowing us to incorporate a standard OCR algorithm into an iterative character classification scheme. The high-level algorithm of the iterative character classification scheme is drawn in Figure 3.

In this context a cluster specifies an area of connected candidate character regions. The algorithm takes into account that

```
FOR EACH cluster in each frame DO
    workCluster = current cluster
    WHILE workCluster not empty DO
        bestClassification = {Char =““
                              errorValue=MAX}
        FOR EACH combination of the regions in the cluster DO
            generate bitmap of size workCluster, where each pixel is set if its corresponding pixel is
                element of the current combination of regions
            FOR EACH connected area in the bitmap DO
                determine character class and error value by standard OCR module.
                store the result in currentClassification.
                IF (currentClassification.errorValue < bestClassification.errorValue) THEN
                    bestClassification = currentClassification
                ENDIF
            END FOR EACH
        END FOR EACH
        IF (bestClassification.errorValue < threshold) THEN
            store recognized character in result list
            remove all regions belonging to the recognized character and all directly connected
                regions from workCluster
        ELSE
            remove all regions in workCluster.
            set bestClassification.Char = ““
        ENDIF
    END WHILE
END FOR EACH
output list of recognized characters (result list)
```

Figure 3: High-level algorithm of the iterative character classification scheme.

there is no way other than testing to identify character regions. However, in reality we do not have to build all combinations but can reduce the number possible by two reasonable heuristics:
For a valid region combination

- at least one region must exceed a minimum expansion, and
- the color spread of the regions must not exceed a maximum Euclidean distance.

The first heuristic is based on the fact that at least the main body of a character is segmented into larger monochrome regions by the Split-and-Merge algorithm. The second heuristic is based on the observation that a single character rarely consists of regions of very differing colors, e.g. such as pink and blue. Thus, by requiring in a valid combination a minimum closeness of the color values of the regions, the number of possible region combinations decreases considerably. In practice, the maximum Euclidean distance of the color values in a valid region combination should be considerably high, thus prohibiting only combinations of regions of considerable different colors.

For optical character recognition we use a feature vector approach as described in [13] and briefly reviewed in the following. A character bitmap is divided into nine segments (Figure 4a). In each segment the number of pixels is determined which belong to one of the four direction classes described by the sixteen 2x2 masks in Figure 4b: horizontal (H), vertical (V), left transverse (L), and right transverse (R). This results in a 36-dimensional feature vector. The vector is normalized and compared to those of the characters in the reference database. It is classified by the nearest neighbor algorithm [1]. The reference database has been trained by 12 twelve different fonts.

This OCR algorithm is far from perfect in comparison to commercial software packages. However, is can be easily integrated into our iterative character classification algorithm.

The recognition result can obviously be improved by taking advantage of the multiple instances of the same text over consecutive frames. Each character of the text often appears somewhat altered from frame to frame due to noise, and changes in background and/or position. We have to detect corresponding character candidate regions in consecutive frames and combine their recognition results into one final character result. However, as we will see in the next section, this step is not needed for our indexing scheme.

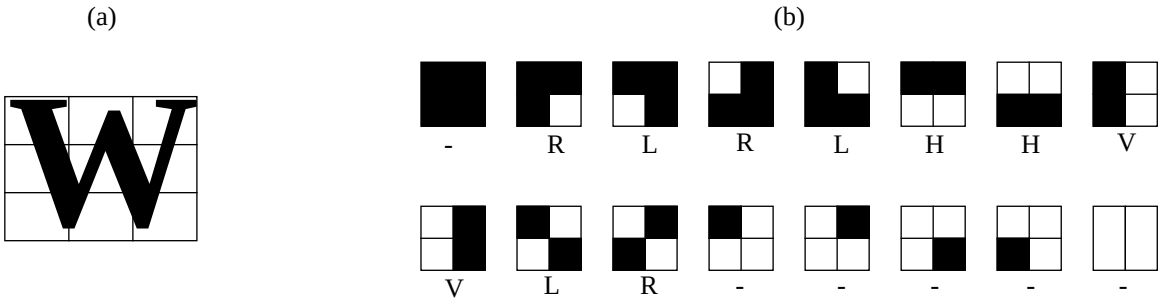

Figure 4: Calculation of feature vectors for optical character recognition as described in [13]:
(a) the division of each character into nine segments.
(b) the sixteen direction elements.

4 INDEXING AND RETRIEVAL

In the preceding chapter we have presented our text recognition algorithms. The upcoming question now is how to use the text recognition result for indexing and retrieval of digital videos. A related question with significant impact on the answer to the original question is what minimal text recognition quality should we assume/demand?
Numerous different font families in all sizes, and sometimes even artistic fonts, are used in artificial text in digital videos. Therefore, OCR errors are very likely. In addition, artificial text generally appears in the foreground of scenes and other "noisy" background which cannot be removed completely by our text segmentation algorithms. Thus, the recognized text will consist of some OCR errors and plenty of garbage characters generated by the mistranslation of background objects. Consequently, our indexing and retrieval scheme should deal well with a poor recognition quality.

Indexing

The indexing scheme is quite simple. Each video sample is processed by the text recognition software. Then, for each frame the recognized characters are stored after deletion of all text lines with fewer than 3 characters. The reason for this deletion is that as experience shows, text lines with up to two characters are produced mainly by background objects and, even if not, consist of semantically weak words such as "a", "by", "in", "to". A sample video frame and the recognized text is given in Figure 5.

Retrieval

Video sequences are retrieved by specifying a search string. Two search modes are supported:

- exact substring matching and
- approximate substring matching.

Original video frame

Recognized text

DAVID
BAMBER
CRQSPIN
BONHAMCITLR
ANVNA
ACHANCELLOF
SUSVNNAH
HARKLI
BARBARA
LIIGHHTIPQT

Figure 5: A sample frame of a pre-title sequence and the recognized text.

Exact substring matching returns all frames with substrings in the recognized text that are identical to the search string. Approximate substring matching tolerates a certain number of character differences between the search string and the recognized text. For approximate substring matching we use the Levenshtein distance L(A,B) between two strings A and B. It is defined as the minimal number of substitutions, deletions and insertions of characters to transform A into B [17]. For each frame we calculate the minimal Levenshtein distance between search string A and all substrings B in the recognized text T [9]. If the minimal distance is below a certain threshold, the appearance of the string in the frame is assumed. Since it can be expected that long words are more likely to have erroneous characters, the threshold value should depend on the length of the search string A.
For instance, if a user is interested in commercials from Chrysler, he/she uses “Chrysler” as the search string and specifies that he/she wants to allow up to one erroneous character per four characters, i.e. he/she wants to allow one edit operation (character deletion, insertion, or substitution) to convert the search string “Chrysler” into some substring of recognized text.

The retrieval user interface is depicted in Figure 6. In the “OCR Query Window” the user formulates his/her query. The result is presented in the “Query Result Window” as a series of small pictures. Multiple hits within one second are grouped into one picture. A single click on a picture displays the frame in full resolution, while a double click starts the external video browser.

5 EXPERIMENTAL RESULTS

In this chapter we discuss two things: Firstly, the performance of our text segmentation and recognition algorithms, and secondly their suitability for indexing and retrieval. Since text is used differently in different film parts and/or film genres, both issues are dealt with separately for three exemplary video genres:

- feature films (i.e. pre-title sequences, credit titles and closing sequences with title and credits),

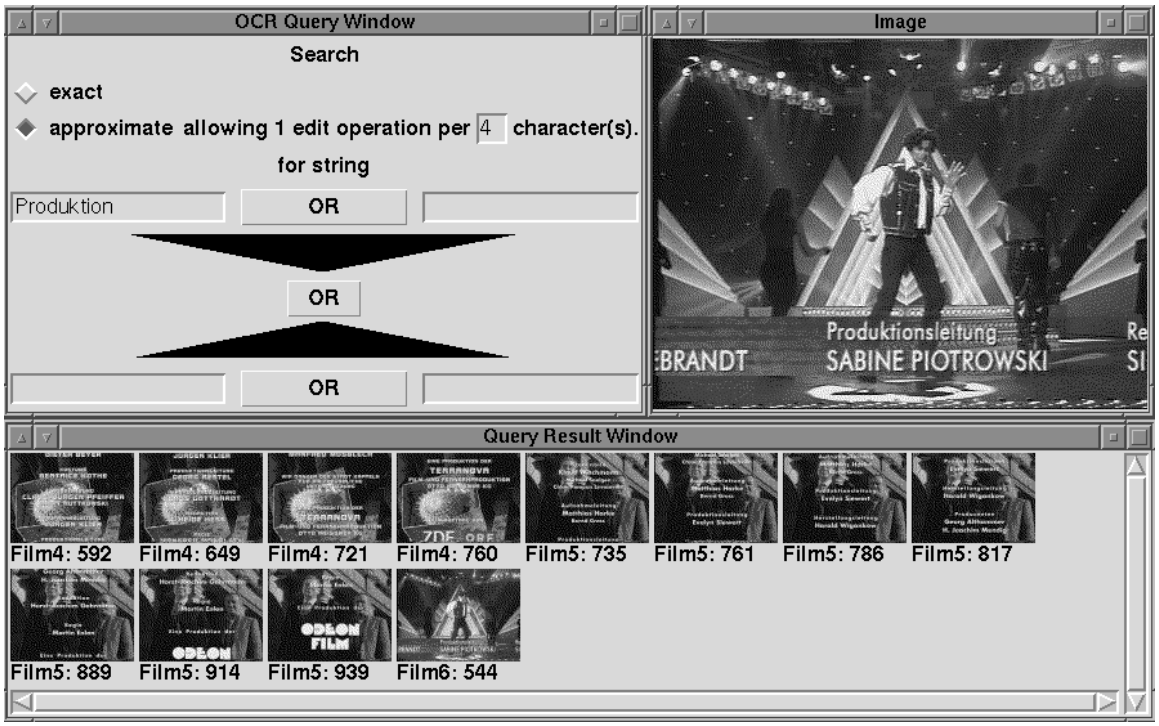

Figure 6: Retrieval user interface

- commercials, and
- newscasts.

Ten video samples for each class have been recorded, adding up to 22 minutes of video. They were digitized from several German and international TV broadcasts as 24-Bit JPEG images at a compression ratio of 1:8, a size of 384 by 288 pixels and at 25 fps. All JPEG images were decoded into 24-bit RGB images.

5.1 Performance of Text Recognition

Segmentation

Before processing each video sample with our text segmentation algorithms we manually wrote down the text appearing in the samples and the frame number range of its visibility. Then we processed each class of video samples with our segmentation algorithms and investigated whether or not a character had been segmented. To be more precise: we measured the quality of our segmentation with regard to the main objective not to discard character pixels. The results for each video genre are averaged and summarized in Table 1. The segmentation performance is very high, ranging from 96% to 99%. Segmentation performance is higher for video samples with moving text and/or moving background than for those where the text and background are stationary. In the latter case our algorithms cannot profit from multiple instances of the same text in successive frames, since all instances of the same character have the same background. Moreover, no background regions can be ruled out by motion analysis. Thus, the segmentation performance is lower. Stationary text in front of a stationary scene can often be found in newscasts. Therefore, segmentation performance in newscasts is lower (96%).
The reduction of candidate character pixels is measured by the reduction factor. It specifies the performance of the segmentation algorithms with regard to our secondary objective: the reduction of the number of pixels which have to considered during the recognition process. The amount of reduction has a significant impact on the quality of character recognition and on speed in the successive processing step. The reduction factor is defined as

$$\text{reduction factor}_{avg} = \frac{1}{\text{\# of frames in video}} \cdot \sum_{f \in \text{video}} \frac{\text{\# of pixels in all character candidate regions of frame f}}{\text{\# of pixels in original frame f}}$$

It ranges from 0.046 to 0.098 and thus also demonstrates the high performance of the text segmentation step. More experimental details are given in [6].

	frames	**characters**	**thereof contained in character candidate regions**		**reduction**
title sequences or credit sequences	7372	6460	6423	99%	0.07
commercials	6858	1074	1065	99%	0.02
newscasts	18624	1464	1411	97%	0.04

Table 1: Segmentation results.

Recognition

The text recognition algorithms are evaluated by two ratio measurements:

- the characters recognized correctly to the total number of characters and
- the additional garbage characters to the total number of characters.

We call the ratios *character recognition rate (CRR)* and *garbage character rate (GCR)*, respectively. However, their exact values have to be determined manually on a tedious basis, frame by frame. Thus, we approximate their values by the following formulas, whose values can be calculated automatically from the manually determined values of text appearances in the segmentation experiments and the calculated recognition result.

$$CRR_{avg} = \frac{1}{\text{\# of frames with text}} \cdot \sum_{\langle f \in video \rangle \wedge \langle f \text{ contains text} \rangle} CRR_f$$

$$\text{GCR}_{avg} = \frac{1}{\text{\# of frames with text}} \cdot \sum_{\langle f \in video \rangle \wedge \langle f \text{ contains text} \rangle} GCR_f$$

where

$$CRR_f \approx 1 - \frac{1}{|W_f|} \cdot \sum_{w \in W_f} \frac{min_{t \in \{\text{all substrings of recognized text in f}\}} \cdot \{L(w,t)\}}{\text{\# of characters of word w}}$$

$$GCR_f \approx max\left\{0, \frac{\text{\# of recognized characters in frame f}}{\text{\# of actual characters in frame f}} - 1\right\}$$

for $\langle f \in video \rangle \wedge \langle f \text{ contains text} \rangle$ and W_f the set of all words actualy appearing in frame f.

Note that the garbage character rate (GCR) is only defined for frames with text occurrences. For frames exhibiting no text we cannot relate the garbage characters to the total number of characters. Thus, we just count their number per text-free frame and call it garbage character count (GCC).

$$\text{GCC}_{avg} = \frac{1}{\text{\# of frames without text}} \cdot \sum_{\langle f \in video \rangle \wedge \langle f \text{ not contains text} \rangle} \text{\# of recognized characters in frame f}$$

The measurements show that the recognition rate is fairly low, around 80% (see Table 2, and for examples Figure 5 and Figure 7). Also, the garbage count is quite high for frames without text, especially for our newscast samples due to their many stationary scenes with stationary text. This observation gives us a strong hint for future research: A computationally cheap detection method for text-free frames has to be developed that can reduce the GCC considerably.

OCR errors and misses originate from the narrowness of our current implementation of the OCR software:

- It is trained with only 12 different fonts.
- It does not deal with touching/merged characters.

Both limitations account for most of the recognition errors. However, as we will see in the following experiment, the performance is sufficient for retrieval purposes. Nevertheless much work must still be done.

.

Figure 7: Two text segmentation and recognition examples, one each for commercial and newscast.

video type	character recognition rate	garbage character rate for frames with text	garbage character count for frames without text
title sequences or credit sequences	0.81	0.27	2.81
commercials	0.80	0.12	7.46
newscasts	0.80	0.89	7.14

Table 2: Recognition results.

5.2 Retrieval Effectiveness

Retrieval effectiveness is the ability of information retrieval systems to retrieve relevant documents while avoiding the retrieval of non-relevant ones. Applied to our domain, we are going to measure the effectiveness of finding all video locations depicting a query word while curbing the retrieval of false locations due to recognition errors or garbage strings generated from candidate character regions belonging to non-characters.

There exist two well-accepted measures for the evaluation of retrieval effectiveness which have been adjusted to our purpose: recall and precision [14]. *Recall* specifies the ratio of the number of relevant video locations found to the total number of relevant video locations in the video database, and *precision* specifies the ratio of the number of relevant retrieval results to the total number of returned video locations. We assume that a video location depicting the search text is retrieved correctly if at least one frame of the frame range has been selected in which the query text appears.

Table 3 depicts the measured average values for recall and precision. They are calculated from the measured values, using each word that occurs once in the video samples as a search string. The recall value for approximate substring matching ranges from 0.5 to 0.79, i.e we get 50% to 79% of relevant material, which is quite high. Also the precision value is considerably high except for the newscasts. Thus, our proposed text segmentation and text recognition algo-

rithms can be effectively used to retrieve relevant video locations. The retrieval application in Figure 6 gives an example.

video type	recall		precision	
	exact substring matching	approximate sub-string matching	exact substring matching	approximate sub-string matching
title sequences or credit sequences	0.56	0.70	0.86	0.72
commercials	0.66	0.79	0.89	0.82
newscasts	0.32	0.50	0.49	0.34

Table 3: Retrieval results.

5.3 Availability

Code for running the algorithms will be available at conference time via FTP from the host *ftp.informatik.uni-mannheim* in the directory */pub/MoCA/*. In addition, readers interested in seeing some of the video clips can retrieve them from *http://eratosthenes.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/MoCA_TextRecognition/*

6 RELATED WORK

Numerous reports have been published about indexing and retrieval of digital video sequences, each concentrating on different aspects. Some employ manual annotation [4][2], others try to generate/compute indices automatically. Automatic indexing generally uses indices based on color, texture, motion, luminance, objects or shape [20], and audio within the video or on external information such as story boards (scripts) and closed captions [7]. Others systems are restricted to specific domains: newscasts [19], football or soccer [3]. None of them try to extract *and* recognize automatically the text appearing in digital videos and use it as an index for retrieval.

Existing work on text recognition has focused primarily on optical character recognition in printed and hand-written documents given the great demand and market for document readers for office automation systems. These systems have attained a high degree of maturity [8]. Further text recognition work can be found in industrial applications, most of which concentrate on a very narrow application field. An example is the automatic recognition of car license plates [16]. The proposed system works only for characters/numbers whose background is mainly monochrome and whose position is restricted.

There exist some proposals regarding text detection and text extraction in complex images and video. In [15], M. Smith and T. Kanade briefly propose a method to detect text in video frames and cut it out. However, they do not deal with the preparation of the detected text for standard optical character recognition software. In particular, they do not try to determine the characters' outline or segment the individual characters. They keep the bitmaps containing text as they are. Human beings have to parse them. They characterize text as a "horizontal rectangular structure of clustered sharp edges" [15] and use this feature to identify text segments. We also employ this feature in our approach, however it plays only a small role in our segmentation process of character candidate regions. Their approach is completely inter-frame and does not utilize the multiple instances under varying conditions over successive frames to enhance segmentation and recognition performance.

B. Yeo and B. Liu propose a caption detection and extraction scheme based on a generalization of a shot boundaries technique for abrupt and gradual transitions to locally restricted areas in the video [18]. According to them, the appearance and disappearance of captions are defined as a localized cut or dissolve. Thus, their approach is inherently intra-frame. It is also very cheap computationally since they operate on compressed MPEG videos. However, captions are only a small subset of text appearances in video. Yeo and Liu's approach seems to fail for general text appearance produced by video title machine, such as scroll titles, since these text appearances cannot just be classified by their sudden appearance and disappearance. In addition, Yeo and Liu do not try to determine the characters' outline, segment the individual characters and translate these bitmaps into text. They propose to use the embedded captions to achieve a content segmentation of news broadcasts in news episodes.

Another interesting approach to text recognition in scene images is that of Jun Ohya, Akio Shio, and Shigeru Akamatsu [10]. Text in scene images exists in 3-D space, so it can be rotated, tilted, slanted, partially hidden, partially shadowed, and it can appear under uncontrolled illumination. In view of the many possible degrees of freedom of text characters, Ohya et al. restricted them to being almost upright, monochrome and not connected, in order to facilitate detection. This makes the approach of Ohya et al. feasible for our aim, despite the fact that they focus on still images rather than on video. Consequently they do not utilize the characteristics typical of text appearing in video. Moreover, we focus on text generated by video title machines rather than on scene text.

7 CONCLUSIONS

We have presented our new approach to text segmentation and text recognition in digital video and demonstrated its suitability for indexing and retrieval. The character recognition algorithm operates on uncompressed frames and make use of intra and inter-frame features of text appearances in digital videos. The algorithm has been tested on title sequences of feature films, newscasts and commercial. The performance of the text segmentation algorithms was always high. Also, recognition performance was high enough to be suitable for our simple indexing scheme. Then, we demon-

strated the usefulness of the recognition result for retrieving relevant video scenes.

Many new applications are conceivable for our automatic text recognition algorithms. For instance, they can be used to find the beginning and end of feature films, since these are framed by title sequences (pre-title and closing sequence). Or they can be used to extract the title of a feature film [11]. In addition, the location of text appearance can be used to enable fast-forward and fast-rewind to interesting parts of the video. This particular feature might be useful in browsing commercials and sportscasts. Specifically, automatic text recognition might be used to find higher semantics in video.

ACKNOWLEDGEMENTS

We thank Oliver Schuster and Frank Stuber for helping us with the implementation of the character segmentation algorithms and Frank Stuber for his implementation of the OCR module. Moreover, many thanks to Christoph Kuhmünch for helping with the experiments.

REFERENCES

1. T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification. *IEEE Trans. Information Theory*, Vol. IT-13, pp. 21-27, January 1967.
2. Marc Davis. Media Streams: Representing Video for Retrieval and Repurposing. *Proc. ACM Multimedia 94*, pp. 478-479, San Francisco, CA, USA, October 15-20, 1994.
3. Yihong Gong, Lim Teck Sin, Chua Hock Chuan, HongJiang Zhang, and Masao Sakauchi. Automatic Parsing of TV Soccer Programs. In *Proc. International Conference of Multimedia Computing and Systems*, pp. 167-174, May 1995.
4. Rune Hjelsvold, Stein Langørgen, Roger Midstraum, Olav Sandstå. Integrated Video Archive Tools. *Proc. ACM Multimedia 95*, San Francisco, CA, Nov. 1995, pp. 283-293.
5. S. L. Horowitz and T. Pavlidis. Picture Segmentation by a Traversal Algorithm. *Comput. Graphics Image Process*. 1, pp. 360-372, 1972.
6. Rainer Lienhart and Frank Stuber. Automatic Text Recognition in Digital Videos. In *Image and Video Processing IV 1996*, Proc. SPIE 2666-20, Jan. 1996.
7. Christopher J. Lindblad, David J. Wetherall, and William Stasior. ViewStation Applications: Implications for Network Traffic. *IEEE Journal on Selected Areas in Communications*, Vol. 13, 1995.
8. Shunji Mori, Ching Y. Suen, Kazuhiko Yamamoto. Historical Review of OCR Research and Development. *Proceedings of the IEEE*, Vol. 80, No. 7, pp. 1029-1058, July 1992.
9. T. Ottmann and P. Widmayer. Algorithms and Data Structures. *BI-Verlag*, Mannheim, 1993. (in German)
10. Jun Ohya, Akio Shio, and Shigeru Akamatsu. Recognizing Characters in Scene Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, pp. 214-220, 1994.
11. Silvia Pfeiffer, Rainer Lienhart, Stephan Fischer, and Wolfgang Effelsberg. Abstracting Digital Movies Automatically. University of Mannheim, Department of Computer Science, Technical Report, TR-96-005, <ftp://ftp.pi4.informatik.uni-mannheim.de/pub/techreports/tr-96-005.ps.gz>, April 1996.
12. Charles A. Poynton. Frequently Asked Questions about Colour. <ftp://ftp.inforamp.net/pub/users/poynton/doc/colour/>, 1995.
13. Alois Regl. Methods of Automatic Character Recognition. Ph. D. thesis, Johannes Kepler University Linz, Wien 1986 (in German).
14. G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. *McGraw Hill*, New York, 1983.
15. Michael A. Smith and Takeo Kanade. Video Skimming for Quick Browsing Based on Audio and Image Characterization. Carnegie Mellon University, Technical Report CMU-CS-95-186, July 1995.
16. M. Takatoo et al., Gray Scale Image Processing Technology Applied to Vehicle License Number Recognition System. in *Proc. Int. Workshop Industrial Applications of Machine Vision and Machine Intelligence*, pp. 76-79, 1987.
17. Wong. Syntactic Image Pattern Recognition. In *Digital Image Processing Methods*, ed. by Edward R. Doughtery, Dekker, New York, 1994.
18. Boon-Lock Yeo and Bede Liu. Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video. in *Digital Video Compression: Algorithms and Technologies*, Proc. SPIE 2668-07 (1996).
19. HongJiang Zhang, Yihong Gong, Stephen W. Smoliar, Shuang Yeo Tan. Automatic Parsing of News Video, *Proc. IEEE Conf. on Multimedia Computing and Systems*, pp. 45-54, 1994.
20. H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution. *Proc. ACM Multimedia 95*, San Francisco, CA, pp. 15-24, Nov. 1995.