**Multimedia Systems Manuscript-Nr.**
(will be inserted by hand later)

# Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions

**Toshio Sato[1] Takeo Kanade[2] Ellen K. Hughes[2] Michael A. Smith[2] Shin'ichi Satoh[3]**

[1] Multimedia Engineering Laboratory, Toshiba Corporation
70 Yanagi-cho, Saiwai-ku, Kawasaki 210-8501, Japan
[2] School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[3] National Center for Science Information Systems (NACSIS)
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-8640, Japan

**Abstract.** The automatic extraction and recognition of news captions and annotations can be of great help locating topics of interest in digital news video libraries. To achieve this goal, we present a technique, called Video OCR (Optical Character Reader), which detects, extracts, and reads text areas in digital video data. In this paper, we address problems, describe the method by which Video OCR operates, and suggest applications for its use in digital news archives. To solve two problems of character recognition for videos, low resolution characters and extremely complex backgrounds, we apply an interpolation filter, multi-frame integration and character extraction filters. Character segmentation is performed by a recognition-based segmentation method, and intermediate character recognition results are used to improve the segmentation. We also include a method for locating text areas using text-like properties and the use of a language-based postprocessing technique to increase word recognition rates. The overall recognition results are satisfactory for use in news indexing. Performing Video OCR on news video and combining its results with other video understanding techniques will improve the overall understanding of the news video content.

**Key words:** Digital video library – Caption – Index – OCR – Image enhancement

## 1 Introduction

Understanding the content of news videos requires the intelligent combination of many technologies including speech recognition, natural language processing, search strategies, and image understanding.

Extracting and reading news captions provides additional information for video understanding. For instance, superimposed captions in news videos annotate the names of people and places, or describe objects.

*Correspondence to*: Toshio Sato(sato@ga2.mmlab.toshiba.co.jp).
Toshio Sato and Shin'ichi Satoh were visiting scientists of the Robotics Institute, Carnegie Mellon University.

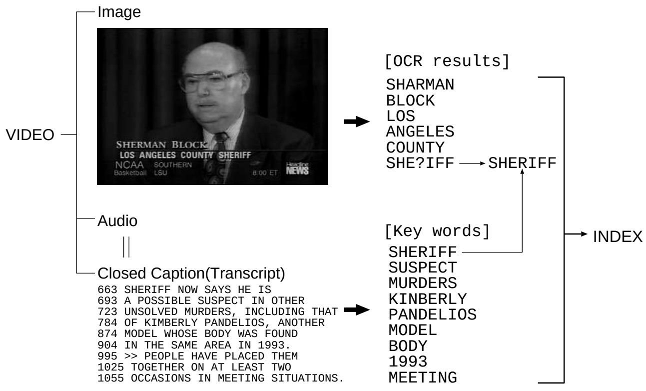


**Fig. 1.** Indexing video data using closed caption and superimposed caption.

Sometimes this information is not present in the audio or cannot be obtained through other video understanding methods. Performing Video OCR on news video and combining its results with other video understanding techniques will improve the overall understanding of the news video content.

For instance, superimposed captions can be used as indices to retrieve desired video data, which are combined with keywords obtained by analysis of closed caption information (See Fig. 1).

Although there is a great need for integrated character recognition in video libraries with text-based queries (Wactlar et al. 1996), we have seen few achievements. Automatic character segmentation is performed for titles and credits in motion picture videos (Smith and Kanade, 1997; Lienhart and Stuber, 1996). While these papers describe methods to detect text regions in videos, insufficient consideration has been given to character recognition.

Character recognition in videos to make indices is described by Lienhart and Stuber (1996) and Kurakake et al.(1997). Lienhart and Stuber (1996) assume that characters are drawn in high contrast against the background to be extracted and have no actual results for recognition. Kurakake et al. (1997) present results for recog-


**Fig. 2.** (a)Examples of caption frame. (b)Magnified image of character($204 \times 14$ pixels). (c)Binary image. (d)Character extraction using a conventional technique.

nition using adaptive thresholding and color segmentation to extract characters. However with news captions, we observe characters which have pixel values similar to those in the background. In such cases, it is difficult to extract characters using either method. Moreover, the character size of the experimental data in both papers is large, while news video requires the reading of significantly smaller characters.

There are similar research fields which concern character recognition in videos (Cui and Huang 1997; Ohya et al. 1994). Cui and Huang (1997) present character extraction from car license plates using Markov random field. Ohya et al. (1994) consider character segmentation and recognition in scene images using adaptive thresholding. While these results are related, character recognition in news videos presents additional difficulties due to low resolution and more severe background complexity.

An example of these issues is illustrated in Fig. 2, which shows one frame from a CNN Headline News broadcast. The video frame contains a typical caption with low resolution characters with heights of ten pixels or less. The background in the example is complicated and in some areas has little difference in hue and brightness from that of the characters. Both words in Fig. 2 have low contrasts against the background so that extraction of characters does not correspond to proper character segments.

The first problem is low resolution of the characters. The size of an image is limited by the number of scan lines defined in the NTSC standard, and video caption characters are small to avoid occluding interesting objects such as people's faces. The size of a character image in the video caption of CNN Headline News is less than $10 \times 10$ pixels. Therefore, the resolution of characters in the video caption is insufficient to implement stable and robust Video OCR systems. This problem can be even more serious if inadequate compression methods such as MPEG are employed.

The problem of resolution is discussed in OCR of World Wide Web images (Zhou et al. 1997) which proposes two recognition techniques for low resolution images: a polynomial surface fitting and the n-tuple methods using color images. However, it would be preferable to improve the quality of images in preprocessing and apply recent huge achievements of recognition for normal resolution characters.

Another problem is the existence of complex backgrounds. Characters superimposed on news videos often have hue and brightness similar to the background, making extraction extremely difficult. However, for many news videos, such conditions are rarely continued in frames in which a superimposed caption appears. In other words, we obtain better conditions using many frames compared with using just one frame, as human eyes can recognize a caption through continuous appearances for a few seconds.

Object detection techniques such as matched filtering are alternative approaches for extracting characters from the background. Brunelli and Poggio (1997) compare several template matching methods to determine which is the best for detecting an eye in a face image. Since every character has a unique shape, simple template matching methods are not adequate for extracting whole character patterns. It is assumed that, for characters composed of line elements, a line detection method (Rowley and Kanade 1994) is more applicable.

We also have a problem with segmenting characters in low quality images. Although there are many approaches for character segmentation (Lu 1995), errors still occur because most methods analyze only a vertical projection profile. Character segmentation based on recognition results (Lee et al. 1996) offers the possibility of improving the accuracy. However, the huge computational cost to select proper segments from combinations of segment candidates is prohibitive and a new approach is needed for practical systems.

In this paper, we present methods to overcome these problems and implement Video OCR for digital news libraries. The methods convert video images into high resolution and high contrast images for recognition of the superimposed captions. We also propose an efficient method to find the best combination for recognition-based segmentation.

The rest of the paper is organized as follows. An overview of the Video OCR system is explained in Section 2. In Sections 3 and 4, image enhancement methods and character extraction methods which cope with both problems are described. In Section 5, character segmentation and recognition methods are described. Section 6 contains results obtained by using Video OCR on news videos. Finally, the importance of Video OCR through applications for digital news archives is discussed in Section 7.

## 2 System Overview

We have designed a Video OCR system which overcomes the low resolution and the complex background problems. The system includes (1)text detection, (2)image enhancement, (3)character extraction, (4)character recognition, and (5)postprocessing. These algorithms are implemented on a workstation and analyze MPEG-1 video data. To evaluate our approach, the system has an alternate processing mode including only text detection, thresholding, simple segmentation, and character recognition. The following sections explain our method in detail.

## 3 Selection and Enhancement of Text Region

### *3.1 Detection*

Since a video news program comprises huge numbers of frames, it is computationally prohibitive to detect each character in every frame. Therefore to increase processing speed, we first roughly detect text regions in groups of frames.

Some known constraints of text regions can reduce the processing costs. A typical text region can be characterized as a horizontal rectangular structure of clustered sharp edges, because characters usually form regions of high contrast against the background.

Smith and Kanade (1997) describe text region detection using the properties of text images. The speed is fast enough to process a $352 \times 242$ image in less than 0.8 seconds using a workstation (MIPS R4400 200MHz).

We first apply a $3 \times 3$ horizontal differential filter to the entire image with appropriate binary thresholding for extraction of vertical edge features. Smoothing filters are then used to eliminate extraneous fragments, and to connect character sections that may have been detached. Individual regions are identified by cluster detection and their bounding rectangles are computed. Clusters with bounding regions that satisfy the following constraints are selected: (1)Cluster size should be larger than 70 pixels, (2)Cluster fill factor should be larger than 45 percent, (3) Horizontal-vertical aspect ratio must be larger than 0.75. A cluster's bounding region must have a large horizontal-to-vertical aspect ratio for satisfying various limits in height and width. The fill factor of the region should be high to ensure dense clusters. The cluster size should also be relatively large to avoid small fragments.

To utilize this method, we select frames and extract regions that contain textual information from the selected frames. If a bounding region which is detected by the horizontal differential filtering technique satisfies size, fill factor and horizontal-vertical aspect ratio constraints, that region is selected for recognition as a text region.

Detection results are selected by their location to extract specific captions which appear at lower positions in frames.

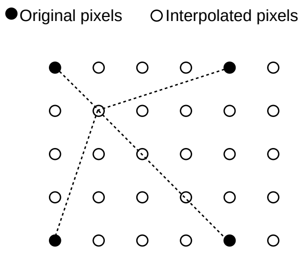


**Fig. 3.** Sub-pixel interpolation (The case of interpolation factor 4 is illustrated).

### *3.2 Improvement of Image Quality*

In television news videos, the predominant difficulties in performing Video OCR on captions are due to low resolution characters and widely varying complex backgrounds. To address both of these problems, we have developed a technique which sequentially filters the caption during frames where it is present. This technique initially increases the resolution of each caption through a magnifying sub-pixel interpolation method. The second part of this technique reduces the variation in the background by minimizing (or maximizing) pixel values across all frames containing the caption. The resulting text areas have good resolution and greatly reduced background variability.

#### 3.2.1 Sub-pixel Interpolation

To obtain higher resolution images, we expand the low resolution text regions by applying a sub-pixel interpolation technique. To magnify the text area in each frame by four times in both directions, each pixel of the original image $I(x, y)$ is placed at every fourth pixel in both $x$ and $y$ directions to obtain the four times resolution image $L(x, y) : L(4x, 4y) = I(x, y)$. Other pixels are interpolated by a linear function using neighbor pixel values of the original image weighted by distances as follows:

$$L(x, y) = \frac{\sum_{(x_0,y_0)\in N(x,y)} d(x - x_0, y - y_0) \cdot I(\frac{x_0}{4}, \frac{y_0}{4})}{\sum_{(x_0,y_0)\in N(x,y)} d(x - x_0, y - y_0)}$$

where $N(x, y) = \{(x_0, y_0) \mid x_0 \in \{\lfloor \frac{x}{4} \rfloor \cdot 4, \lceil \frac{x}{4} \rceil \cdot 4\}, y_0 \in \{\lfloor \frac{y}{4} \rfloor \cdot 4, \lceil \frac{y}{4} \rceil \cdot 4\}\}$ and $d(x, y) = \|(x, y)\|^{-1}$ (Also see Fig. 3).

#### 3.2.2 Multi-frame Integration

For the problem of complex backgrounds, an image enhancement method by multi-frame integration is employed using the enhanced resolution interpolation frames. Although complex backgrounds usually have movement, the position of video captions is relatively stable across frames. Furthermore, we assume that captions have high intensity values such as white pixels. Therefore, we employ a technique to minimize the variation of the background by using a time-based minimum pixel value search. (For black characters, the same technique could


**Fig. 5.** Result of sub-pixel interpolation and multi-frame integration (upper). Binary image (lower). Pixel size is 813 × 56.


**Fig. 4.** Improving image quality by multi-frame integration.

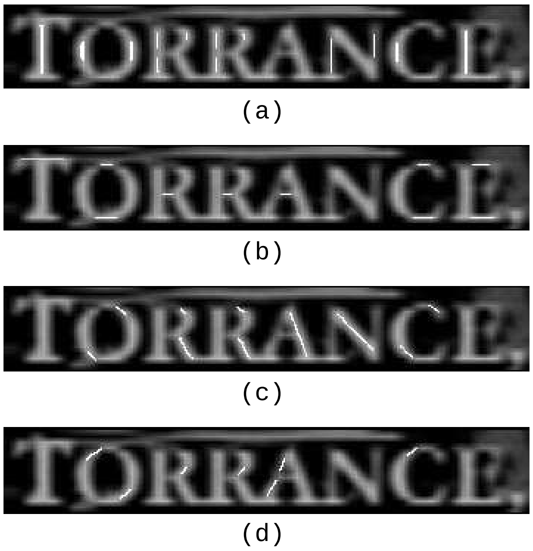


**Fig. 6.** Learning data of filters (white pixels). (a) 0° . (b)90°. (c)−45°. (d)45°.

be employed using a time-based maximum search.) With this technique, an enhanced image is made from the minimum pixel value that occurs in each location during the frames containing the caption. By taking advantage of the video motion of non-caption areas, this technique results in text areas with less complex backgrounds while maintaining the existing character resolution. (See Fig. 4).

The sub-pixel interpolated frames, $L_i(x,y), L_{i+1}(x,y), \ldots, L_{i+n}(x,y)$ are enhanced as $L_m(x,y)$ via

$$L_m(x,y) = min(L_i(x,y), L_{i+1}(x,y), \ldots, L_{i+n}(x,y))$$

where $(x,y)$ indicates the position of a pixel and $i$ and $i+n$ are the beginning frame number and the end frame number, respectively. These frame numbers are determined by text region detection (Smith and Kanade 1997).

An example of effects of both the sub-pixel interpolation and the multi-frame integration is shown in Fig. 5. The original image in Fig. 2, which has less than $10 \times 10$ pixel size for each character, is enhanced to have a size of approximately $30 \times 40$ pixel size by this method. Characters have smooth edges in contrast with notches of the original image in Fig. 2.

## 4 Extraction of Characters

### *4.1 Character Extraction Filter*

To further reduce the effect of complex backgrounds, a specialized filter based on correlation is used. We recognize that a character consists of four different directional line elements: 0° (vertical), 90° (horizontal), −45° (anti-diagonal), and 45° (diagonal). We employ a filter which integrates the output of four filters corresponding to those line elements.

To make learning data for filters, we select from an actual television frame, caption pixels which correspond to 0°, 90°, −45°, and 45° line elements, shown as white pixels in Fig. 6.

The size of the filter is defined to include only a line element of characters. In this paper, we consider $15 \times 3$, $3 \times 7$, $9 \times 7$ and $9 \times 7$ filters to detect 0°, 90°, −45°, and 45° line elements, respectively. Values of the filters are determined by averaging pixels for each position of the neighboring area according to the learning data. Each filter image is normalized to have an average value of zero. All filter values are multiplied by a common fixed coefficient to make the output pixel values remain in 8-bit data.

At the time of processing, a preprocessed image $L_m(x,y)$ is filtered by calculating correlation with the filters $F_i(x,y)$, where $i = \{0,1,2,3\}$, to extract each line element. Values of $F_i(x,y)$ are fixed as shown in Fig. 7 and applied to all other images.

$$L_{f,i}(x,y) = \sum_{y_0=-h_i}^{h_i} \sum_{x_0=-w_i}^{w_i} L_m(x+x_0, y+y_0) \cdot F_i(x_0+w_i, y_0+h_i)$$

where $w_i$ and $h_i$ represent the area of the filters.

Positive values at the same position among filtered images are added to integrate four directional line elements.

$$L_f(x,y) = \sum_{i=0}^{3} L'_{f,i}(x,y)$$


**Fig. 8.** Result of character extraction filter. (a)0°. (b)90°. (c)−45°. (d)45°. (e)Integration of four filters. (f)Binary image.

**Fig. 10.** Edges of peak in vertical projection profile. White lines indicate candidates of character segments.

**Fig. 11.** Result of segmentation. Correct segments are selected based on recognition.

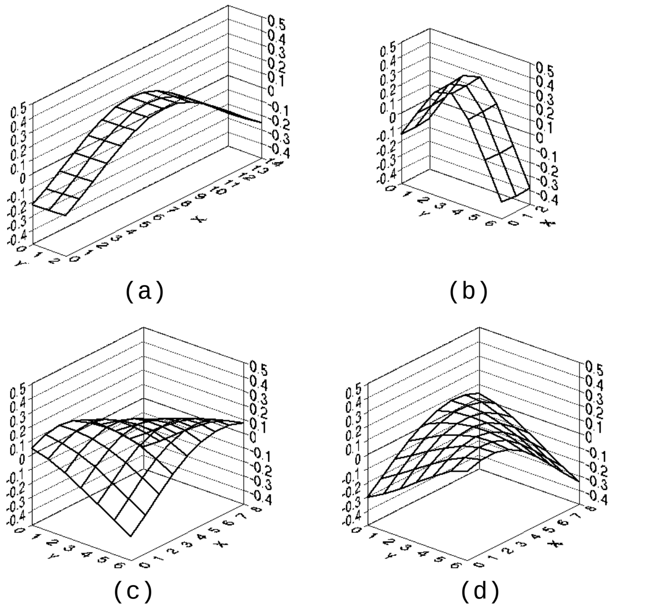


**Fig. 7.** Character extraction filters $F_i(x, y)$. (a)0°. (b)90°. (c)−45°. (d)45°.

$$L'_{f,i} = \begin{cases} 0 & \text{if } L_{f,i}(x,y) \le 0 \\ L_{f,i}(x,y) & \text{otherwise} \end{cases}$$

Pixels for characters have high values, so thresholding at a fixed value $\theta$ is applied to reduce noise for the final output of the character extraction filter $L_{filter}$.

$$L_{filter} = \begin{cases} 0 & \text{if } I(x,y) \le \theta \\ L_f(x,y) & \text{otherwise} \end{cases}$$

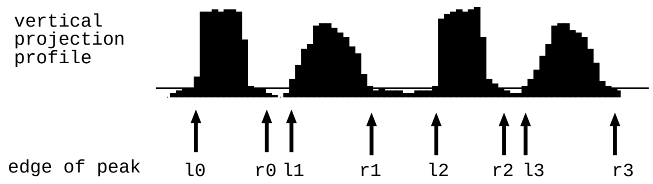


**Fig. 9.** Edge of peak in vertical projection profile. $l_i$ and $r_i$ are left edges and right edges of peaks, respectively.

Results of the character extraction filter are shown in Fig. 8. Each filter emphasizes pixels which correspond to a line direction not only for the left word which includes learning data but also for the right word. The integration of four directional filtered images shown in Fig. 8(e) indicates that the filter extracts characters from complex backgrounds; these backgrounds are not eliminated by the simple thresholding method shown in Fig. 5.

While we acquired training data by using Roman font characters to design character extraction filters, the filters also work well with Gothic font character images. Although the filter output appears weaker at intersections of strokes, it does not present much of a problem in segmenting and recognizing characters.

### *4.2 Thresholding/Projection*

Thresholding at a fixed value for the output of the character extraction filter $L_{filter}$ produces a binary image which is used to determine positions of characters and recognize characters.

An example of the thresholding is shown in Fig. 8(f). This result also explains why the complex background which appears in Fig. 5 is eliminated. Horizontal and vertical projection profiles of the binary image are used to determine candidates for character segmentation.

### *4.3 Simple Segmentation*

As we search for white characters, peaks in a vertical projection profile indicate boundaries between characters. As shown in Fig. 9, a position at which the projection value changes from low to high compared with a fixed threshold value is detected as a left edge of the peak. In the same manner, a right edge of the peak is detected as high to low transition. The left and right edge positions of the peak correspond to the left and right boundaries of a character segment candidate, respectively.

Spurious valleys in the projection profile sometimes cause a character to be segmented into two or more pieces as shown in Fig. 10. White lines in Fig. 10 illustrate detected edges of peaks in a vertical projection profile. Eight characters out of 18 are over-segmented, although the edges include proper character segments.

Since the edges of the peaks include correct segment boundaries of the character, we will devise a method for selecting the best combination of edges of the peaks to find character segments. The method integrates the edge positions with the character recognition results, as described in the next section.

Top and bottom boundaries of character segments are simply determined by detecting beginning and end positions in a horizontal projection profile.

## 5 Integration of Recognition and Segmentation

### *5.1 Character Recognition*

We use a conventional pattern matching technique to recognize characters. An extracted character segment image is normalized in size and converted into a blurred gray scale image by counting the number of neighbor pixels. This preprocessing makes the recognition robust to changes in thickness and position. The normalized gray scale data $n(x,y)$ are matched with reference patterns $ref_c(x,y)$ based on a correlation metric. The matching metric $m_c$ is described as follows:

$$m_c = \frac{\sum n(x,y) \cdot ref_c(x,y)}{\sqrt{\sum (n(x,y))^2}\sqrt{\sum (ref_c(x,y))^2}} \tag{1}$$

A category $c$ which has the largest $m_c$ in reference patterns is selected as the first candidate. In the same manner, second and third candidates are selected. The reference patterns $ref_c(x,y)$ are built by averaging sample data.

### *5.2 Selecting Segmentation Result by Character Recognition*

Segment candidates which are detected by the simple segmentation method may include over-segmented characters as described in Section 4.3.

To select a correct pair of edges which represents a character segment, correspondences between character segments and their similarities $m_c$ are evaluated.

The peaks of the vertical projection $\boldsymbol{P}$ consist of pairs of left edge $l_i$ and right edge $r_i$.

$$\boldsymbol{P} = \{(l_i, r_i) \mid i = 0, 1, 2, \ldots, M\}$$

where $M+1$ is the number of peaks.

A candidate of character segmentation $\boldsymbol{C_j}$ is defined as a series of segments $(l_b, r_e)$, where $l_b$ and $r_e$ are left and right edges that appear in $\boldsymbol{P}$:

$$\boldsymbol{C_j} = \{(l_0, r_{\alpha_j}), (l_{\alpha_j+1}, r_{\beta_j}), \ldots, (l_{\gamma_j+1}, r_M)\}$$

where $0 \le \alpha_j < \beta_j < \ldots < \gamma_j + 1 \le M$.

A segmentation is evaluated on how well it can result in a series of readable characters. That is, for a segmentation $\boldsymbol{C_j}$, an evaluation value $E_v(\boldsymbol{C_j})$ is defined as the sum of the maximum similarities for each segment normalized by the number of segments $N_{um}(\boldsymbol{C_j})$.

$$E_v(\boldsymbol{C_j}) = \sum_{(l_b, r_e) \in \boldsymbol{C_j}} h(l_b, r_e) / N_{um}(\boldsymbol{C_j})$$

where $h(l_b, r_e)$ is the maximum similarity of a segment between $l_b$ and $r_e$ among the reference patterns $ref_i(x,y)$ in Eq. (1).

The best character segmentation $\boldsymbol{C}$ is the one with the largest evaluation value.

$$\boldsymbol{C} = \arg \max_{\boldsymbol{C_j} \in \boldsymbol{P}} (E_v(\boldsymbol{C_j})) \tag{2}$$

The search for the best segmentation is performed efficiently with the constraint of the character width to reduce the calculation cost.

We now describe the detailed procedure of the search method for determining the best segmentation $\boldsymbol{C}$:

$$\boldsymbol{C} = \{(l_0, r_{s(1)}), (l_{s(1)+1}, r_{s(2)}), \ldots, (l_{s(n)+1}, r_M)\}$$

where $n \le M$.

We process two consecutive segments at a time; we consider the first segment of the two to be correct and fixed if both the first and the second segments have high similarities. This process is repeated to fix all segments. As the first segment starts from the first left edge $l_0$, the end point of the first segment is determined as $r_{s(1)}$ via

$$s(1) = \arg \max_{u=\{0,\ldots,M-1\}} \{h(l_0, r_u) + h(l_{u+1}, r_v)\} \tag{3}$$

where $r_u - l_0 < w_c, r_v - l_{u+1} < w_c$ and $w_c$ is the maximum width of a character. Then, the start point of the next segment is $l_{u+1}$.

We continue to determine the end points of segments $r_{s(k+1)}$ step by step.

$$s(k+1) = \arg \max_{u=\{s(k)+1,\ldots,M-1\}} \{h(l_{s(k)+1}, r_u) + h(l_{u+1}, r_v)\} \quad (4)$$

where $r_u - l_{s(k)+1} < w_c, r_v - r_{u+1} < w_c$.

If $v$ reaches $M$, the process terminates and both segments are determined to be results. Further, if $(l_{u+1} - r_u)$ in Eq. (4) exceeds a fixed value, the $r_u$ is considered to be the end of a word and the process starts from Eq. (3) for the next word.

This method enables us to determine the best segmentation $\boldsymbol{C}$ faster than by calculating the evaluation value for all combinations of segments in Eq. (2).

The edges of the peaks which are detected in Fig. 10 are selected by character recognition to segment characters shown in Fig. 11.

## 6 Experimental Results

### *6.1 Evaluation of Video OCR*

We evaluate our method using CNN Headline News broadcast programs, received and digitized. Video data is encoded by MPEG-1 format at $352 \times 242$ resolution. We use seven 30-minute programs, which include 256 kinds of caption frames, 375 lines and 965 words. Specific superimposed captions which appear at the same position in a frame, such as name and title captions in Fig. 1, are selected for the evaluation.

Table 1 shows results of text detection. The text detection described in Section 3.1 detects a frame segment which consists of sequential frames including the same captions. The process also detects text regions in a frame which correspond to lines in the text region. Words are segmented if distances between adjacent character segments exceed a certain value as described in Section 5. The word detection is also affected by text region detection errors.

Table 1: Result of text detection

| | Correct | Total |
|---|---|---|
| Frame segment | 235 (91.8%) | 256 |
| Text region | 336 (89.6%) | 375 |
| Words | 733 (76.0%) | 965 |

In the seven 30-minute videos, the detected text regions include 5,076 characters, which consist of 2,370 Roman font characters and 2,706 Gothic font characters. Using these detected characters, we evaluate our method with respect to character recognition rates. Our method includes the sub-pixel interpolation, the multi-frame integration, the character extraction filter and the segmentation results selection by character recognition which are described in Sections 3, 4, and 5. Using a workstation (MIPS R4400 200MHz), it takes about 120 seconds to process a caption frame block and about 2 hours to process a 30-minute CNN Headline News program.

Table 2 shows character recognition rates for Video OCR using seven 30-minute CNN Headline News videos. The percentage of correct Roman characters (76.2%) is lower than that of Gothic characters (89.8%) because Roman font characters have thin line elements and tend to become scratchy. The total recognition rate is 83.5%.

Table 2: Character recognition of Video OCR

| | Roman | Gothic | Roman+ Gothic |
|---|---|---|---|
| Correct characters | 1807 | 2430 | 4237 |
| Total characters | 2370 | 2706 | 5076 |
| Recognition rate | 76.2% | 89.8% | 83.5% |

To compare our results with those from a conventional OCR approach, we implement a simple OCR program. It consists of binarization of an image by straightforward thresholding at a fixed value (no sub-pixel interpolation, no multi-frame integration, and no character extraction filter), character extraction by horizontal and vertical projections of the binary image (no integrating with character recognition), and matching by correlation (the same algorithm as ours).

Fig. 12 shows examples of recognition results for our method and the conventional OCR. According to these results, it is obvious that our approaches, especially the image enhancement and the character extraction filter, improve character recognition of video data, thus solving the resolution and the complex background problems.

Table 3 shows character recognition results of the conventional OCR. The recognition rate is 46.5%, which is almost half of our results and less than half of an average commercial OCR rate for documents (Information Science Research Institute 1994). The recognition rate of Roman font characters is lower because of their thin lines which correspond to one or less pixel in the binary image.

Table 3: Character recognition of conventional OCR

| | Roman | Gothic | Roman+ Gothic |
|---|---|---|---|
| Correct characters | 921 | 1439 | 2360 |
| Total characters | 2370 | 2706 | 5076 |
| Recognition rate | 38.9% | 53.2% | 46.5% |

### *6.2 Postprocessing for Word Recognition*

To acquire text information for content-based access of video databases, high word recognition rates for Video OCR are required. The Video OCR recognizes only 54.8% (402 out of 733 words), even though the total recognition rate of characters is 83.5%.

We apply postprocessing, which evaluates differences between recognition results with words in the dictionary, and selects a word having the least differences. The differences are measured among three candidates of the recognition result weighted by the similarity $m_c$. We use two kinds of dictionaries: the *Oxford Advanced Learner's Dictionary* (Oxford University Computing Services 1997) (69,517 words) and word collections which are compiled by analyzing closed caption information of videos (4,824 words).

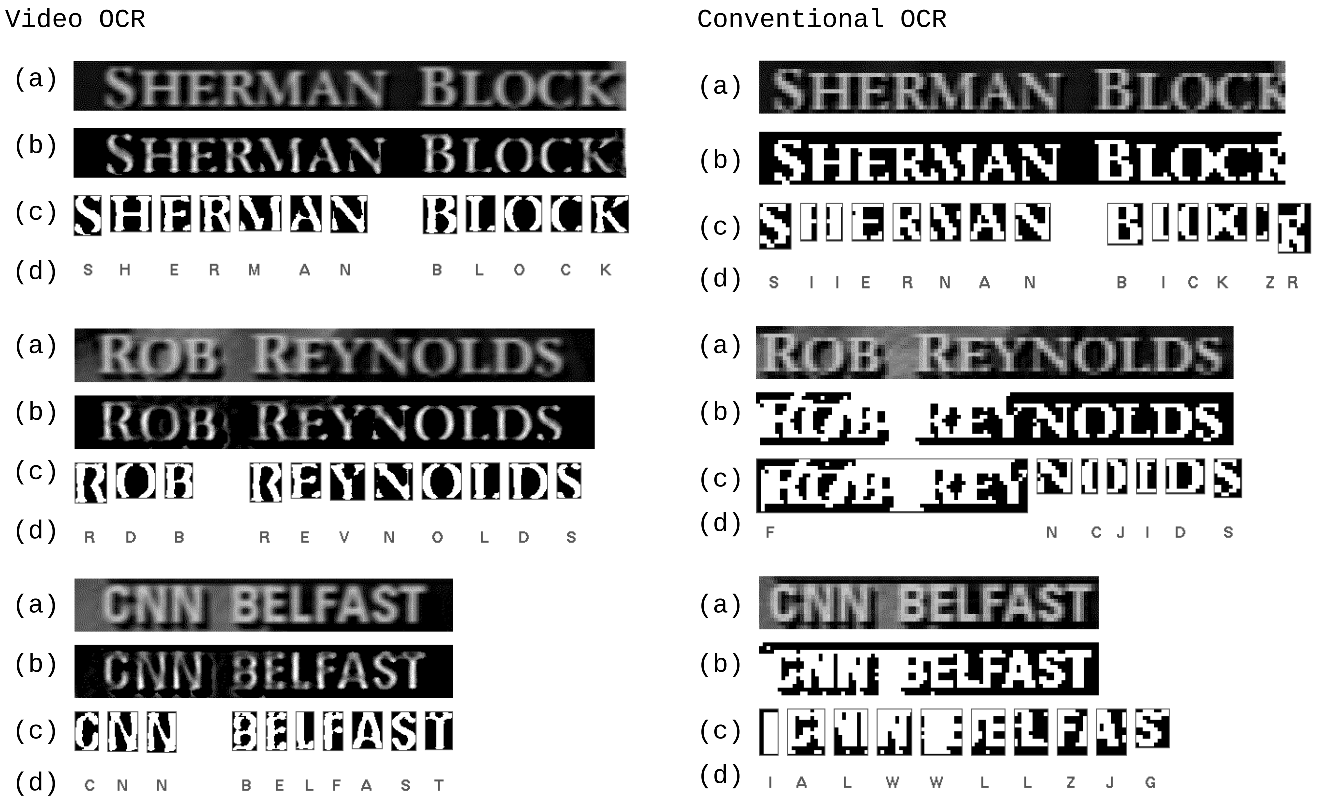


**Fig. 12.** Result of Video OCR and conventional OCR. (a)Original or enhanced image. (b)Binary image. (c)Segmentation Result. (d)The first candidates of recognition.

To match video caption recognition results with data in the dictionaries, we define the similarity between them. This similarity is defined based on the edit distance (Hall and Dowling 1980).

Assume that $x$ is the character recognition result of a word. Let $x(i)$ be the $i$-th letter of the recognition result $x$. Assuming that $p$ is any one letter of a recognition result (e.g., $x(i)$), we define $p_j^c$ and $p_j^s$ as the resultant character and its score of the letter of the $j$-th precedence, respectively; e.g., $p_1^c$ is the first preceded character of the letter $p$. $x(i)_1^c$ is the first preceded character of the letter. Likewise, let $y$ be a word to be matched with, and $y(i)$ be the $i$-th letter of $y$. In addition, $x(i..)$ $(y(i..))$ denotes the substring of $x$ $(y)$ begins at $i$-th position. The modified edit distance $d_c(x, y)$ is defined recursively as follows:

$$d_c(x, y) = min \left\{ \begin{array}{l} 1 + d_c(x(2..), y), \\ 1 + d_c(x, y(2..)), \\ c_c(x(1), y(1)) + d_c(x(2..), y(2..)) \end{array} \right\}$$

$$d_c(\varepsilon, \varepsilon) = 0$$

where $\varepsilon$ represents a null string. $c_c(p, q)$ is the cost function between characters $p$ and $q$, which is defined as follows:

$$c_c(p, q) = \begin{cases} 1 & (\forall_i p_i^c \neq q) \\ 1 - \frac{p_i^s}{p_1^s} & (p_i^c = q) \end{cases}$$

The distance is calculated using the dynamic programming algorithm.

Then, the normalized distance $\hat{d_c}(x, y)$ between $x$ and $y$ is defined as follows:

$$\hat{d_c}(x, y) = \frac{d_c(x, y)}{max(len(x), len(y))}$$

where $len(x)$ returns the length of the string $x$. When $x$ and $y$ are the same, the distance is 0, whereas when $x$ and $y$ are totally different (i.e., $x$ and $y$ do not share any character), the distance is 1.

As shown in Table 4, both dictionaries improve word recognition; 131 words using the Oxford dictionary and 162 words using the closed caption dictionary are corrected. However, 89 correctly recognized words using the Oxford dictionary and 107 recognized words using the closed caption dictionary are missed since each dictionary does not include corresponding words.

Postprocessing using a combined dictionary increases the word recognition rate up to 70.1%. The combined dictionary results maintain the total number of corrected words with decreasing the number of misses by more than half.

Table 4: Word Recognition rate with postprocessing

(Total: detected 733 words)

| | Correct words(rate) | Corrected by post. | Missed by post. |
|---|---|---|---|
| w/o Post. | 402 (54.8%) | — | — |
| Post.(Ox) | 442 (60.3%) | 131 | 89 |
| Post.(CC) | 455 (62.1%) | 162 | 107 |
| Post.(Ox+CC) | 514 (70.1%) | 157 | 43 |

Ox: Oxford dictionary, CC: Closed caption dictionary

Table 5 shows vocabularies of each dictionary. The Oxford dictionary lacks place names of small cities ("Augusta"), common peoples' names ("Kawakita") and some particular proper nouns ("CNN" and "Whitewater"). The closed caption dictionary is missing place names ("Georgia"), usual people's name ("Ian") and general words ("courtesy" and "midnight"). It is considered that the Oxford dictionary and the closed caption dictionary mainly confirm general words and words for people, organizations or place names, respectively. If we use much closed caption data, words that appear in the Oxford dictionary may be included in the closed caption dictionary.

Table 5: Vocabularies of dictionary data (* included)

| Word | Oxford | Closed caption |
|---|---|---|
| WASHINGTON | * | * |
| GEORGIA | * | |
| AUGUSTA | | * |
| AUSTIN | | |
| LYNNE | * | * |
| IAN | * | |
| KAWAKITA | | * |
| GINGRITCH | | * |
| CNN | | * |
| NBA | | * |
| WHITEWATER | | * |
| COURTESY | * | |
| MIDNIGHT | * | |

### *6.3 Performance of Indexing News Video*

To determine the benefits of Video OCR, we evaluated the results as a method to obtain indices for news videos.

First, we calculate redundancy between superimposed captions and closed caption information for all words and recognized results using seven 30-minute CNN videos. Table 6 shows the results, which indicates that 56.1 % of the superimposed caption words are not present in the closed caption words and the rate is almost same for the recognized captions. According to the result, Video OCR will improve retrieval abilities of news videos because almost half of the information obtained from the video caption can be added as new index data.

Table 6: Overlap of video captions

| | Words in video caption | Words not in closed caption | Rate |
|---|---|---|---|
| Total | 965 | 541 | 56.1% |
| Video OCRed | 514 | 261 | 50.8% |

Table 7 shows overall performance of the word recognition and the retrieval of new indices by the Video OCR. The word recognition rate for superimposed captions is 53.3 % which results in 27.0 % of the superimposed caption words becoming new indices.

Table 7: Performance of Video OCR

| Total words | Detected words | OCRed words | Words not in closed caption |
|---|---|---|---|
| 965 | 733(76.0%) | 514(53.3%) | 261(27.0%) |

### *6.4 Evaluation of Video Preprocessing with Commercial OCR*

To further evaluate the value of the video preprocessing for text regions, we performed the character recognition after video text preprocessing using a standard commercially available OCR. We used the video text area preprocessing described in Sections 3 through 4.2 to process the text and inverted the result so that it appeared just as black text would appear on white paper. Commercial OCRs are specially trained to recognize black text with a white background, so our video text area preprocessing enabled us to use the preprocessed text areas as the input to the commercial OCR.

We used 7 days of CNN World View news programming, which is about 6 hours of video. After using the text detection as described in Section 3.1, these videos were processed in two different ways for comparison: (1) text areas were extracted and binarized using a simple thresholding technique; (2) text areas were preprocessed and extracted using the methods of Sections 3.2 through 4.2 (with the combined sub-pixel interpolation and multi-frame integration). Both sets of results were fed into the commercial OCR. No postprocessing was employed at this time. The results for news word recognition of detected text areas given in Table 8 illustrate the strength of the video text area preprocessing techniques.

Table 8: Word Recognition Rate for News Programming

| | Total Words | Correct Words | % Correct |
|---|---|---|---|
| No Preprocessing | 1420 | 540 | 38.0 |
| Preprocessing | 1420 | 1000 | 70.4 |

It should be noted that the CNN World View news programming contains a variety of different types of news captions and different character fonts and sizes. Even though there is a large variety of character types in this data set, the results show that good recognition rates can be attained. In addition, although we did not score the results, text areas in commercials were also processed and recognized. These video text area preprocessing techniques work well on a variety of character fonts and sizes to convert the complex backgrounds and poor resolution characters into character data which standard


(a) (b) (c)

**Fig. 13.** Examples of correctly recognized captions with words that are not in the audio. (a)Name and title. (b)Detailed Description (c)Commercial.

OCR packages can correctly recognize. Some example results for these are given in the following section.

The combination of video text area preprocessing with a commercial OCR enables our system to take advantage of the text enhancements of the video preprocessing while increasing the number and sizes of recognizable fonts. While no postprocessing error correction was used in this evaluation, application of the postprocessing techniques presented in Section 6.2 would further improve the word recognition rate. The overall results for approximately 6 hours of video attaining 70% correct word recognition show that the preprocessing was so successful in increasing resolution and eliminating background noise, that even a commercial OCR could be used with our system to obtain good results.

## 7 Applications of Video OCR

In Section 6, we showed that our approach improves retrieval abilities of news videos and that Video OCR results bring new index data which is not included in closed captions.

In this section, to expand on the types of knowledge in which Video OCR can improve news video understanding, a few examples are presented for the many different types of news captioning that can aid in video searching: identification of person or place, name of news-worthy event, date of event, stock market and other news statistics, and news summaries. Using these examples, we discuss applications of Video OCR based on the actual usage of the superimposed caption.

### *7.1 Name and Title*

An example in Fig. 13(a) illustrates captioning with the identification of the person and his or her title. The name and title of the person shown in this example is not spoken in the audio nor is it present in the closed caption information. To recognize this kind of frame using Video OCR, name and title information in the image are obtained. This name and title information is valuable in helping to match people's faces to their names (Satoh and Kanade 1997).

Even if the video caption information is included in the closed caption, the results are valuable for associating image information with audio or closed caption information. Nakamura and Kanade (1997) introduce a semantic association method between images and closed caption sentences using a dynamic programming technique based on segments of the topics in the news programs. Using Video OCR results, the association accuracy will be improved since we can put more milestones in a segment of the topic.

### *7.2 Detailed Description*

Fig. 13(b) shows an example of news summary captioning. During the appearance of the news summary caption, the audio contains only music and the closed caption has no data. The diversely changing video behind the news summary text exploits the strengths of our video preprocessing techniques. The summary given in the example was recognized with no errors. Without accurate Video OCR, this type of news content understanding would be lost.

### *7.3 Detection of Commercial Segments*

We also found that Video OCR can potentially be helpful in identifying commercials. An example shown in Fig. 13(c) is a correctly detected and read text area from a commercial that was broadcast during the news. Since it is currently difficult to identify advertisements within news, Video OCR may prove to be a valuable tool in that endeavor.

## 8 Conclusions

Accurate Video OCR can provide unique information as to the content of video news data, but poses some challenging technical problems. We addressed the preprocessing problems of low resolution data and complex backgrounds by combining sub-pixel interpolation on individual frames and multi-frame integration across time. We also evaluated new techniques for character extraction and segmentation using specialized filtering and recognition-based character segmentation. Our approach, especially video preprocessing, makes the recognition rate double that of conventional methods. Application of postprocessing techniques gives further improved accuracy. Overall word recognition rates exceed

50% for superimposed captions, and 25% of the captions can be used as new indices which are not included in closed captions.

The information gained by performing Video OCR is often unobtainable from other video understanding techniques. Accurate Video OCR is valuable not only for conventional video libraries with text-based searching but also for new types of video content understanding currently not possible, such as: matching faces to names, associating different types of information, identifying advertisements, tracking financial or other statistics across time, and capturing the content of a sequence of video described only by music and captions.

## Acknowledgments

This work has been partially supported by the National Science Foundation under grant No. IRI-9411299. The authors would like to thank Yuichi Nakamura for valuable discussions and for providing postprocessing data.

## References

1. Wactlar HD, Kanade T, Smith MA, Stevens SM (1996) Intelligent access to digital video: The Informedia project. IEEE Computer 29: 46-52
2. Smith MA, Kanade T (1997) Video skimming and characterization through the combination of image and language understanding technique. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 775-781
3. Lienhart R, Stuber F (1996) Automatic text recognition in digital videos. Proceedings of SPIE Image and Video Processing IV 2666: 180-188
4. Kurakake S, Kuwano H, Odaka K (1997) Recognition and visual feature matching of text region in video for conceptual indexing. Proceedings of SPIE Storage and Retrieval in Image and Video Databases 3022: 368-379
5. Cui Y, Huang Q (1997) Character extraction of license plates from video. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 502-507
6. Ohya J, Shio A, Akamatsu S (1994) Recognizing characters in scene images. IEEE Trans Pattern Analysis and Machine Intelligence 16: 214-220
7. Zhou J, Lopresti D, Lei Z (1997) OCR for World Wide Web images. Proceedings of SPIE Document Recognition IV 3027: 58-66
8. Wu V, Manmatha R, Riseman EM (1997) Finding text in images. Proceedings of the second ACM International Conference on Digital Libraries, Philadelphia, PA, ACM Press, New York, NY, pp. 3-12
9. Brunelli R, Poggio T (1997) Template matching: Matched spatial filters and beyond. Pattern Recognition 30: 751-768
10. Rowley HA, Kanade T (1994) Reconstructing 3-D blood vessel shapes from multiple X-ray images. AAAI Workshop on Computer Vision for Medical Image Processing, San Francisco, CA
11. Lu Y (1995) Machine printed character segmentation - an overview. Pattern Recognition 28: 67-80
12. Lee SW, Lee DJ, Park HS (1996) A new methodology for gray-scale character segmentation and recognition. IEEE Trans Pattern Analysis and Machine Intelligence 18: 1045-1050
13. Information Science Research Institute (1994) 1994 annual research report. http://www.isri.unlv.edu/info/publications/anreps.html
14. Oxford university computing services (1997) The Oxford text archive. http://ota.ox.ac.uk/
15. Hall PAV, Dowling GR (1980) Approximate string matching. ACM Computing Surveys 12: 381-402
16. Satoh S, Kanade T (1997) NAME-IT: Association of face and name in video. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 368-373
17. Nakamura Y., Kanade T (1997) Semantic analysis for video contents extraction - spotting by association in news video. Proceedings of the fifth ACM International Multimedia Conference, Seattle, WA, pp.393-401

Toshio Sato received his BS degree in Fine Measurement Engineering and his MS degree in Systems Engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 1985 and 1987, respectively. He joined Toshiba Corporation, Kawasaki, Japan, in 1987 and currently is a specialist in the Multimedia Engineering Laboratory. He was a visiting scientist at Carnegie Mellon University, Pittsburgh, Pennsylvania, from 1996 to 1998 and worked for the Informedia Digital Video Library Project. His main research interests include object detection, pattern recognition, and image and video understanding. He is a member of the IEEE.

Takeo Kanade received his Doctoral degree in Electrical Engineering from Kyoto University, Japan, in 1974. After holding a faculty position at Department of Information Science, Kyoto University, he joined Carnegie Mellon University in 1980, where he is currently U. A. Helen Whitaker University Professor of Computer Science and Director of the Robotics Institute. Dr. Kanade has worked in multiple areas of robotics: computer vision, manipulators, autonomous mobile robots, and sensors. He has written more than 200 technical papers and reports in these areas, as well as more than ten patents. He has been the principal investigator of several major vision and robotics projects at Carnegie Mellon. Dr. Kanade has been elected to the National Academy of Engineering. He is a Fellow of the IEEE, a Founding Fellow of American Association of Artificial Intelligence, and the founding editor of International Journal of Computer Vision. He has received several awards including the Joseph Engelberger Award, JARA Award, Otto Franc Award, Yokogawa Prize, and Marr Prize Award. Dr. Kanade has served for government, industry, and university advisory or consultant committees, including Aeronautics and Space Engineering Board (ASEB) of National Research Council, NASA's Advanced Technology Advisory Committee and Advisory Board of Canadian Institute for Advanced Research.

Ellen K. Hughes received her B.S. degree in Electrical Engineering from Wright State University in Dayton, Ohio in 1986, and her M.S. degree in Electrical and Computer Engineering from Carnegie Mellon University in 1990. From 1987 through 1997, she worked at the Westinghouse Science and Technology Center, acquired by Northrop Grumman in 1996. There she pursued research and development in signal and image processing on a wide variety of applications from biomedical technology to sonar image processing to automated machine vision systems. In 1997 she joined the Informedia Digital Video Library Project Team at Carnegie Mellon University. She is a member of Tau Beta Pi and the IEEE.

Michael A. Smith received his Ph.D. in Electrical and Computer Engineering at Carnegie Mellon University in 1997. His dissertation topic was the Integration of Image, Audio and Language Understanding for Video Characterization and Variable Rate Skimming. He received his B.S. in Electrical Engineering from North Carolina A&T State University in 1991, and his M.S. in Electrical Engineering from Stanford University in 1992. He is a member of IEEE as well as Eta Kappa Nu, Tau Beta Pi, and Pi Mu Epsilon. He has published papers in the areas of pattern recognition, biomedical imaging, video characterization, and interactive computer systems. His current interest are image classification and recognition, and content based video understanding. His research is an active component of the Informedia Digital Video Library project at Carnegie Mellon.

Shin'ichi Satoh received his BS, MS, and Ph.D. from the University of Tokyo, Tokyo, Japan in 1987, 1989, and 1992 respectively. He is an Associate Professor since 1998, and was a Research Associate from 1992 to 1998, in the Research and Development Division, National Center for Science Information Systems, Tokyo, Japan. He was a visiting researcher in the Robotics Institute, Carnegie Mellon University, Pittsburgh, U.S.A., from 1995 to 1997, and worked for the Informedia Digital Video Library Project. His research interests include multimedia database, multimodal video analysis, and video understanding.

This article was processed by the author using the LaTeX style file *cljour2* from Springer-Verlag.