JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 7, No. 4, December, pp. 336–344, 1996
ARTICLE NO. 0029

Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database

HAE-KWANG KIM*

Institut de Recherche en Informatique de Toulouse, Universite Paul Sabatier, 118, route de Narbonne 31062, Toulouse Cedex, France

Received April 18, 1996; accepted September 6, 1996

An efficient automatic text detection and location method for video documents is proposed and its application for the content-based retrieval of video is presented and discussed. Target frames are selected at fixed time intervals from shots detected by a scene-change detection method. For each selected frame, segmentation by color clustering is performed around color peaks using a color histogram. For each color plane, text-lines are detected using heuristics, and the temporal and spatial position and the text-image of each text-line are stored in a database. Experimental results for text detection in video images and the performance of the method are reported for various video documents. A user interface for text-image based browsing is designed for direct content-based access to video documents, and other applications are discussed. © 1996 Academic Press

I. INTRODUCTION

Ergonomic retrieval methods and user interfaces based on appropriate structuring and indexing schemes are essential for digital video applications such as video editing, video libraries, etc. Video uses two different channels for conveying a story to the spectator: the visual channel carrying text, graphics, and moving image (or sequence of images) and the auditory channel carrying music, speech, and artificial or environmental sounds. Video contents, regardless of the channel, can be divided into two categories: (1) perceptual contents such as colors, shapes, textures, frequencies, timbers, and their temporal changes and (2) semantic contents such as objects and events and their relations. In general, perceptual contents are easier to analyze automatically, and semantic contents are easier to manipulate linguistically. Thus, techniques for inferring semantically meaningful data from perceptual contents analysis are particularly useful.

The state of the art of signal processing, pattern recognition and its applications to image, and sound analysis is such that most research work on automatic recognition of semantic audiovisual objects for indexing and retrieval of image and video documents has used relatively low-level perceptual content analysis such as texture (Picard and Minka, 1995), shape (Sclaroff and Pentland, 1995; Tegolo, 1994), and color (Sakamoto *et al.*, 1994) in image data, and various sound parameters for audio (Blum *et al.*, 1995). Automatic or semi-automatic methods are favored, for the sake of excluding subjectivism and saving the labor cost and tedious repetitive work associated with human indexing. It turns out that some important semantic clues can be recovered directly from the perceptual contents.

Recently, researchers have tried to propose methods for semantic speech or text-based retrieval of video. Some researchers have proposed the textual indexing of nontextual documents based on analyzing neighboring textual parts with IR technology (Agosti *et al.*, 1995). A simplified case-based reasoning system derived from AI is applied to video taking a video clip as a case in (Gordon and Domeshek, 1995). When coded close captions are available, word spotting techniques can be used to find keywords in the textual transcription of dialogue and narration (Hauptmann and Smith, 1995). The same researchers also propose to use speech recognition techniques to directly produce the transcription. Query by text retrieval has also been proposed for speech database by automatically indexing speech segment with key phonetic sequences using speech recognition techniques in (Schauble and Wechsier, 1995). Automatic text-based video structuring methods based on caption data are proposed in (Shahraray and Gibbon, 1995).

The ability to extract existent linguistic information like text-images in video is helpful for content-based access to video. For example, video can be browsed through text-images of the name of actors and directors in films and of statistics in sports programs. In this article, a text location algorithm for indexing and structuring of video documents and its application is presented and discussed. In Section II, related works on text region locating are reviewed and then the text location algorithm is explained in Section III. A user interface for text-image-based browsing is pre-

* E-mail: kim@irit.fr.

sented for content-based retrieval of video documents and other applications are discussed in Section IV. Experimentation results are given with discussion on the performance of the algorithm in Section V, and the article is concluded in Section VI.

II. RELATED WORK ON TEXT REGION DETECTION

Text region detection methods have been developed for enhancing computing efficiency by guiding costly OCR algorithms using statistics (Kuo and Agazzi, 1994) or shape analysis (Rocha and Pavlidis, 1995) to reduced zones in textual document processing. The text zone detection method is applied to binarized documents and exploits global heuristic features of the specific target document. For example, text lines can be simply distinguished from blank lines by counting foreground pixels on each horizontal line in the case of the current paragraph of this paper. If the count is zero, the line belongs to a blank line and otherwise to a text line. Detected text zones can be macroscopic, such as columns in a newspaper, and further segmented into smaller zones, such as text lines, and then into character zones (Lu, 1995).

The above text region detection by global heuristics does not work for documents such as maps or engineering drawings due to the high composition flexibility and presence of many textual and graphical objects. For this type of documents, connected components are obtained from the binarized image. Text strings are located by some heuristic geometric composition rules of connected components (Lai and Kasturi, 1994) or by searching linear alignments of connected components applying Hough transform to the center positions of the connected components (Fletcher and Kasturil, 1988).

A method of character recognition in video scene images under no controlled conditions for robotic applications is proposed (Ohya *et al.*, 1994). The binarized image is obtained by applying a local thresholding technique on a source grey-level image. Connected components are obtained from the binarized image and character segment candidate components are selected by investigating the difference in average grey-levels of the background and the foreground inside the enclosing box of the connected component. Characters are confirmed by character recognition technology. This method has some fundamental problems for video document analysis applications: binary segmentation is not appropriate for video images which consist of various objects of different grey-levels; a character segment can be split into several connected components due to various inherent video noises; the whole process is very time consuming.

New methods of automatic text region detection have been published recently, aiming at application to image databases (Zhong *et al.*, 1995). The target text lines are presumed to be in a specific direction (for experimentation, horizontal text lines are detected).

—*Connected component method.* Dominant peaks are calculated from RGB color histogram and other colors are merged into the nearest peaks. Connected components of homogeneous color are obtained and candidate characters are selected using heuristics restricting character size and the number of aligned characters for a text-line. Text-lines are obtained as sets of horizontally aligned detected characters.

—*Spatial variance method.* This method is based on the assumption that the spatial variance of the background is lower than the text zones. A variance image is obtained by calculating the variance value within a horizontal window (1×21 pixels) for each pixel position in a source grey-level image. An edge operator (Canny edge detector) is applied to the variance image. Horizontal edges are detected and merged. Two edge lines are paired as the upper and the lower lines of a text bounding box. When more precise location of the enclosing box is needed, connected component analysis is added (hybrid method).

The connected component method is not appropriate for video documents because it is based on the effectiveness of the segmentation method which guarantees that a character is segmented as one connected component separated from other objects. It is generally very difficult to have such a segmentation method for low-resolution video images with various inherent noises. The spatial variance method is an interesting approach for the text detection of video images, and thus it is implemented and tested for comparison with the algorithm proposed in this article. The experimentation is done mostly on CD cover images with all methods and on some car scene images from video with the spatial variance method.

III. TEXT REGION LOCATION FOR VIDEO IMAGES

An automatic text detection and location algorithm for video documents is explained in detail in this section. The algorithm should comply with the characteristics of video documents such as low-image resolution, strong noise, high-composition complexity of various objects, and uncontrolled conditions. Low-pass filtering for noise reduction is not applied because it can cause problems on frequently present characters of one-pixel thickness in video images. The computing should be fast due to the enormous quantity of video data. The algorithm should deal with various characters of different styles and sizes present in video documents. False positive detection is more encouraged than false negative detection because automatic 100% correct detection is hardly achievable in this condition so that human intervening cost for the correction—explained in Section IV—should be considered.

The process of the algorithm is drawn in Fig. 1. Target

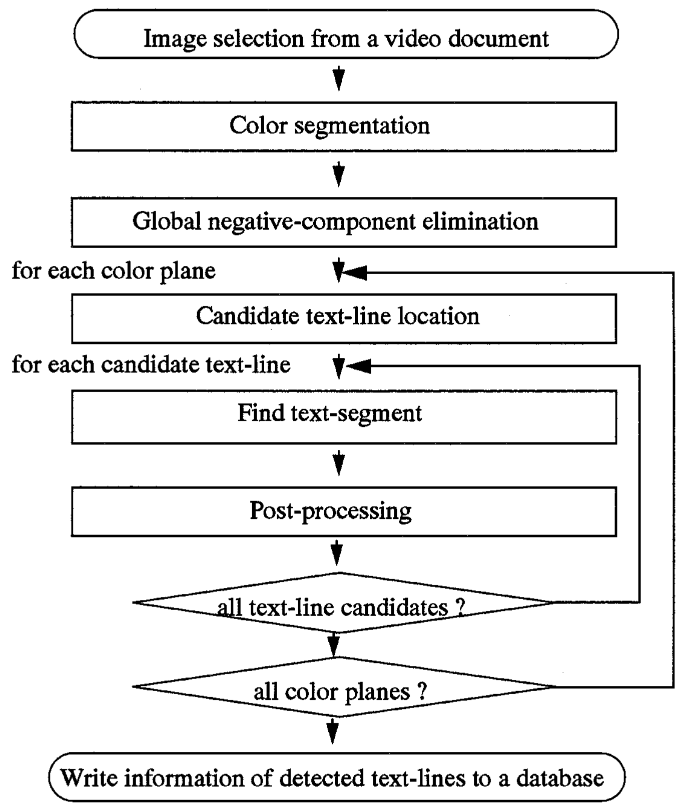
Image selection from a video document
Color segmentation
Global negative-component elimination
for each color plane
Candidate text-line location
for each candidate text-line
Find text-segment
Post-processing
all text-line candidates ?
all color planes ?
Write information of detected text-lines to a database

FIG. 1. Process of the text detection algorithm.

images are selected at fixed time intervals from shots detected by a scene-change method. Each image is segmented by clustering colors around dominant colors and global negative shape components are marked as eliminated. Text-line detection is executed on each color plane, regarding the selected color as the foreground and the rest of colors as the background. Candidate text-lines of an arbitrary direction (horizontal direction is used for the experimentation because it's the most prevailing direction of text in video documents) are located by eliminating local negative shape components and using Y-signature in the color plane. Text-segments are detected in each candidate text-line based on heuristics. Post processing is applied for having the final text-line location by merging or rejecting the detected text-segments. The information of the detected text-lines such as enclosing box position and size and text-image are stored in a database along with the temporal position of the selected image. The result of each step of the text detection for an example image is shown in Fig. 2.

Some heuristics used are: (1) text-lines are separated from the image borders; (2) text-lines are horizontally aligned characters of the same color; (3) the color of text-line is well contrasted from the background colors; and (4) the width and height of characters are within the maximum and the minimum limits. Other heuristics are presented when they appear in the explanation of the algorithm in the following paragraphs. These heuristics do not distinguish character sets so that the algorithm can detect text lines of international character sets. The proposed method has some limits set from the heuristics: it cannot detect text-lines of varying colors or with visual effects and cannot detect arbitrary direction of text-lines. But, detection of horizontal homogeneously colored text results in an already very efficient capture of semantically meaningful elements for automatic video indexing and structuring. Each step of the algorithm is presented in detail in the following subsections.

1. *Image Selection*

The visual channel of video documents consists of large number of sequential images with temporal redundancy so that key-image selection methods have been proposed for abstracted representation of video documents on the basis of "shot" unit. The shot (a sequence of images without temporal interruptions) is a fundamental unit in video shooting and editing processes. Automatic shot change detection methods have been proposed by detecting cuts and transition effects (Nagasaka and Tanaka, 1991; Zhang *et al.*, 1993; Aigrain-Joly, 1994) and shots are further segmented into microsegments with homogeneous camera motion by a spatio-temporal image analysis method (Joly and Kim, 1996). Key-image selection methods have been proposed for a shot such as a simple method of predetermined temporal positions (first, second, last, etc.) and nonlinear sampling methods of local minima of motion (Wolf, 1995) and iterative image selection by thresholding the dissimilarity value with the previous image (Yeung and Liu, 1995).

There are two kinds of text-images in video documents: scenery text-images taken by the camera, which move on the screen by motions of the camera and of objects carrying text-images, and edited text-images, most of which appear at fixed screen positions at an arbitrary temporal position for a sufficient duration for human perception in a shot regardless of any background scene changes. The occupation ratio of edited text-images to the whole image is frequently so small that for the above nonlinear key-image selection methods, the threshold value needs to be set quite low not to miss right frames, leading to very sensitive selection of images with small camera and object motions. Edited text-images usually last on the screen during a sufficient time *Ts*, required for the perception of the viewer and scenery text-images usually last much longer so that selection of key-images at the fixed time intervals of *Ts* can work well for the text-image detection of video documents. *Ts* of 2 s is regarded as appropriate from observation with the selection rate of 1:60 for NTSC video.

2. *Color Segmentation*

Segmentation is a very important and critical preprocessing step in general vision and pattern recognition appli-

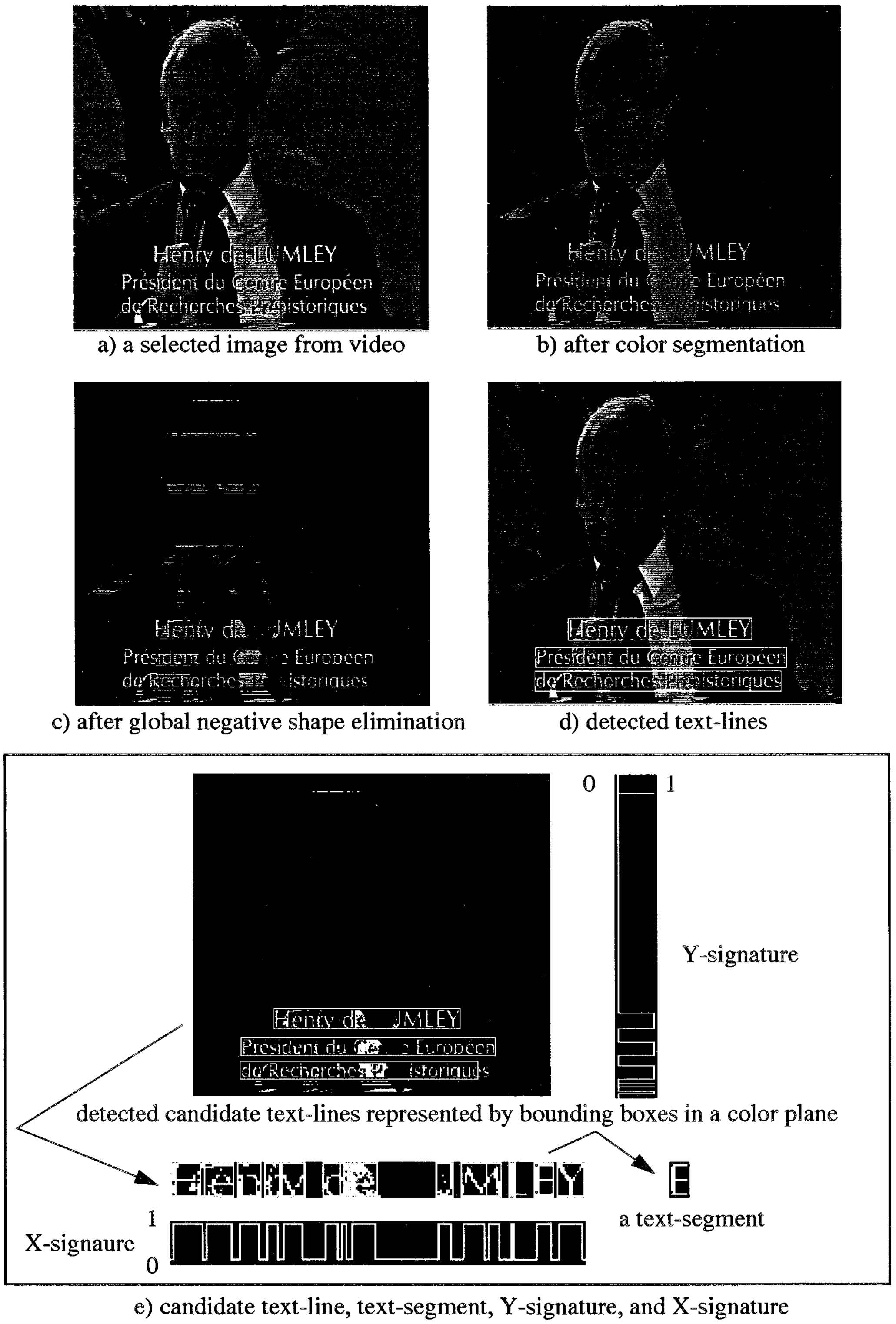
Henry de LUMLEY
Président du Centre Européen
de Recherches Préhistoriques
a) a selected image from video
b) after color segmentation
c) after global negative shape elimination
d) detected text-lines
0
1
Y-signature
detected candidate text-lines represented by bounding boxes in a color plane
1
X-signaure
0
a text-segment
e) candidate text-line, text-segment, Y-signature, and X-signature

FIG. 2. Images showing each step of and terms used in the paper (the source image is extracted from a video FAUST '94, directed by Maryse Baute and Veronique Dardoize, © Mairie de Toulouse, Jan. 1995).

cations. Hundreds of methods have been reported with various techniques based on thresholding, clustering or edge detection with statistics, fuzzy logic technology, and neural networks for grey value or multi-spectral images, but there is no single method which can be considered good for all applications (Pal and Pal, 1993). Local adaptive bi-level thresholding methods (Taxt *et al.*, 1989) are effective and efficient for automatic application to textual documents, maps, and engineering drawings in which relevant objects belong to either the foreground or the background. For video documents, a color segmentation method is required because the images are composed of several objects of different colors. Selection of a color coordinate space and its color distance metrics takes an important role for the result of a color segmentation algorithm. Color spaces of $L^*u^*v^*$ and $L^*a^*b^*$ are known as matching to human color perception experience and *Whitening* (*KL* transform of *RGB*) color space is reported as satisfactory from an experimentation of an iterative clustering color segmentation method (Uchiyama *et al.*, 1994), but the computing time is important for the color space transform and color distance calculation. *RGB* space is used and the color distance $d(c1, c2)$ between two colors $c1(r1, g1, b1)$ and $c2(r2, g2, b2)$ is measured for the algorithm as the following formula: $d(c1, c2) = (r2 - r1)^2 + (g2 - g1)^2 + (b2 - b1)^2$.

```
Algorithm Color_segmentation
Start
—Construct CH, the color histogram for the input image
  While CH is not empty
    —Select the dominant color dc of maximum count in CH
    For each color rc resting in CH
      If the color distance $d(rc, dc) < T_c$
        —Set the color of pixels of rc to dc in the input image
        —Eliminate rc from the color histogram
      EndIf
    EndFor
    —Eliminate dc from CH
  EndWhile
End
```

The threshold value T_c determines the performance of the following steps of the text detection algorithm. With a high threshold value, computing complexity is diminished but different objects can be merged into one object, and with a low threshold value, an object can be split into parts with different colors and more computing time is required. T_c is chosen as the minimum value required for getting sufficient pixels of perceptually homogeneous color characters into one color plane. From experimentation on a sample set of various perceptually significant text-lines of different colors and backgrounds, The T_c is set to 1/25 of the maximum color distance by finding the minimum quantification level of the maximum color distance with sufficient text-line pixels into one color plane. For computing the color histogram *CH*, only the most significant 5 bits are considered for each 8-bit color channel for the computing speed. The least significant 3 bits can be ignored with the T_c value of 1/25. The algorithm detects only sufficiently contrasted characters from the background ignoring the blurring sign of a store in a deep field. Figures 2a and 2b show respectively the original image and the transformed image after the segmentation.

3. *Global Negative Shape Elimination*

In this step, all 8-connected homogeneous color components containing border pixels or excessively long horizontal lines are marked as eliminated for the following steps on the basis of the heuristics that characters are separated from other nontextual objects and borders. Various heuristics for nontextual components can be used such as too large or small connected components from calculating connected component enclosing box. For the present work, only very long horizontal line is adopted for nontextual object heuristics for the sake of computing speed. The process is resumed in the following algorithm and the resulting image is shown in Fig. 2c).

```
Algorithm Global_negative_shape_elimination
Start
—Eliminate connected pixels to four boundaries of the image
—Eliminate connected pixels to horizontal lines whose length > $T_l$
End Global_negative_shape_elimination
```

A line-by-line propagation method is implemented for eliminating connected pixels to a given line and T_l is set to 50 pixels long for the experimentation.

4. *Candidate Text-Line Location*

Continuing text detection steps are performed on each color plane. A text-line is assumed to be a horizontally aligned set of characters. For each row of the color plane, its *Y-signature* value is set to 0 when it is decided that it does not constitute a text-line by the counts of line segments and the foreground pixels, and otherwise to 1. Candidate text-lines are obtained as a set of consecutive rows of *Y-signature* set to 1. Text-lines of height smaller than the predetermined minimum text-line height are eliminated. The process is performed as the following algorithm and the result image is shown in Fig. 2d).

```
Algorithm Candidate_text-line_location
Start
  For each row of the color plane, row
    —Calculate cf, the count of foreground pixels
    —Calculate start, end,
      the positions of the first and the last foreground pixels
    —Calculate cl, the count of the line segments of foreground color
    —Calculate fpr, the foreground pixel ratio = $cf/(end - start + 1)$
    If ($cl < T_{cl}$) or ($fpr > T_{fp\_max}$) or ($fpr < T_{fp\_min}$)
      —Set Y-signature value to 0
    Else
      —Set Y-signature value to 1
    EndIf
  EndFor
  —Construct text-lines with consecutive rows of Y-signature set to 1
  —Eliminate text-lines of which height $< T_h$
End
```

T_{cl} defines the minimum character counts which constitute a text line and is set to 3 for the experimentation. The value of T_{fp_max} and T_{fp_min} eliminates rows which are excessively full or vacant and are set to 0.8 and 0.2 each for the experimentation. The T_h value determines the minimum height of a text-line and is set to 7 pixels from the experimentation. Candidate text-lines are shown in Fig 2e).

5. *Find Text-Segment*

For each candidate text-line, local negative shape components such as locally excessive horizontal line segments and filled boxes of foreground color are eliminated with pixels connected to them. For each column of a text-line, the *X-signature* value is set to 0 when the column contains no foreground pixel, and otherwise to 1. Candidate text-segments are obtained as a set of consecutive columns of *X-signature* set to 1. Each text-segment is tested against the text-segment heuristics. If the width of the accepted text-segment is longer than the height, the text-segment is divided into sub-text-segments. If any of the sub-text-segments fails text-segment heuristics, the whole text-segment is rejected. The following algorithm explains the process and Fig. 2e illustrates text-line, text-segment, and *X-signature* for the example image.

```
Algorithm Find_text-segment
Start
  —height = height of the text-line
  —Eliminate line segments (whose length $> T_{ls}$) and connected pixels
  —Eliminate filled boxes (whose width $> T_{fb}$) and connected pixels
  —Calculate X-signature and locate text-segments
  For each text-segment, ts
    If is_text-segment (ts) == True
      If width of ts > height
        —Divided ts into sub-text-segments with width of height
        (the last sub-text-segment has width of height from the end point)
        If is_text-segment (sts) == True for all sub-text-segments, sts
          —Save ts as text-segment
        EndIf
      Else
        — Save ts as text-segment
      EndIf
    EndIf
  EndFor
End
```

```
Boolean Function is_text-segment (text-segment)
Start
  —width = width of text-segment
  —height = height of text-segment
  —area = width * height
  —tpc = total foreground pixel count
  —ipc = isolated foreground pixel count
  —isolated_ratio = ipc / area
  —surface_ratio = tpc / area
  —calculate Y-signature for text-segment after eliminating isolated pixels
  —text_height = maximum count of consecutive rows of Y-signature of 1
  —height_ratio = text_height / height
  —aspect_ratio = height / width
  If (surface_ratio $> T_{s\_max}$) and (aspect_ratio $< T_a$)
    return True
  If (surface_ratio $< T_{s\_min}$) or (surface_ratio $> T_{s\_max}$)
    or (isolated_ratio $> T_{ir}$) or (height_ratio $< T_{hr}$)
    return False
  Else
    return True
  EndIf
End
```

T_{ls} and T_{fb} define the local negative shape component of long line segment and full box and are set respectively to twice and one-third of the text-line height. T_{s_max}, T_{s_min}, T_{ir}, T_{hr}, and T_a determine the text-segment heuristics of maximum and minimum ratio of foreground pixels, maximum ratio of isolated foreground pixels, minimum number of consecutive nonempty rows, and maximum aspect ratio for I-shape characters. $T_{s\_max} = 0.8$, $T_{s\_min} = 0.2$, $T_{ir} = 0.1$, $T_{hr} = 0.6$, $T_a = 0.3$ have been determined from the experimentation as the values with which real text-segment are not rejected.

6. *Post Processing*

Text-segments which are closer than a certain distance are merged into a macro text-segment and if the number of text-segments in the macro text-segment is smaller than the minimum number of characters for a text-line or its width is smaller than a predetermined value, the macro text-segment is rejected. If more than one macro-segment is found, the enclosing box is calculated: the horizontal start and end position are set to the start position of the leftmost text-segment and the end position of the rightmost text-segment (lost text-segments are recovered) and the vertical start and end position are set to the start and end positions of a macro text-segment. The spatial and temporal position and text-image data are stored in a database. The final detected text-lines are shown in Fig. 2d.

```
Algorithm Post_processing
Start
  —Merge text-segments which are closer than $T_d$
  For each merged text-segment
    width = the width of merged text-segment
    count = the number of text-segments that it contains
    If width > $T_w$ or count > $T_{cl}$
      —Save as macro text-segment
    EndIf
  EndFor
  If more than one macro text-segment found
    —(start_x, start_y)(end_x, end_y); the upper-left and the lower-right corner points of the enclosing box
    —start_x = the horizontal start position of the leftmost text-segment
    —end_x = the horizontal end position of the rightmost text-segment
    —start_y = the vertical start position of a macro text-segment
    —end_y = the vertical end position of a macro text-segment
  Endif
  —Save the text-image, the temporal and geometric data of the enclosing box in a database
End
```

T_d for grouping individual characters is set to 20 pixels and T_w for detecting multi-character text-segments is set to 50 pixels and T_{cl} is set to 3 in the candidate text-line location step.

IV. APPLICATIONS

Video programs convey very important information related to the content in text-images such as bibliographic information (titles, names of actors, producers, etc.) for all sorts of programs, statistics in sports games, product names in commercials, locations of accident in news reports, etc., so that detected text-images can be very useful semantic index for video content retrieval. The text-image browser in Fig. 3 is designed for text-image based video browsing.

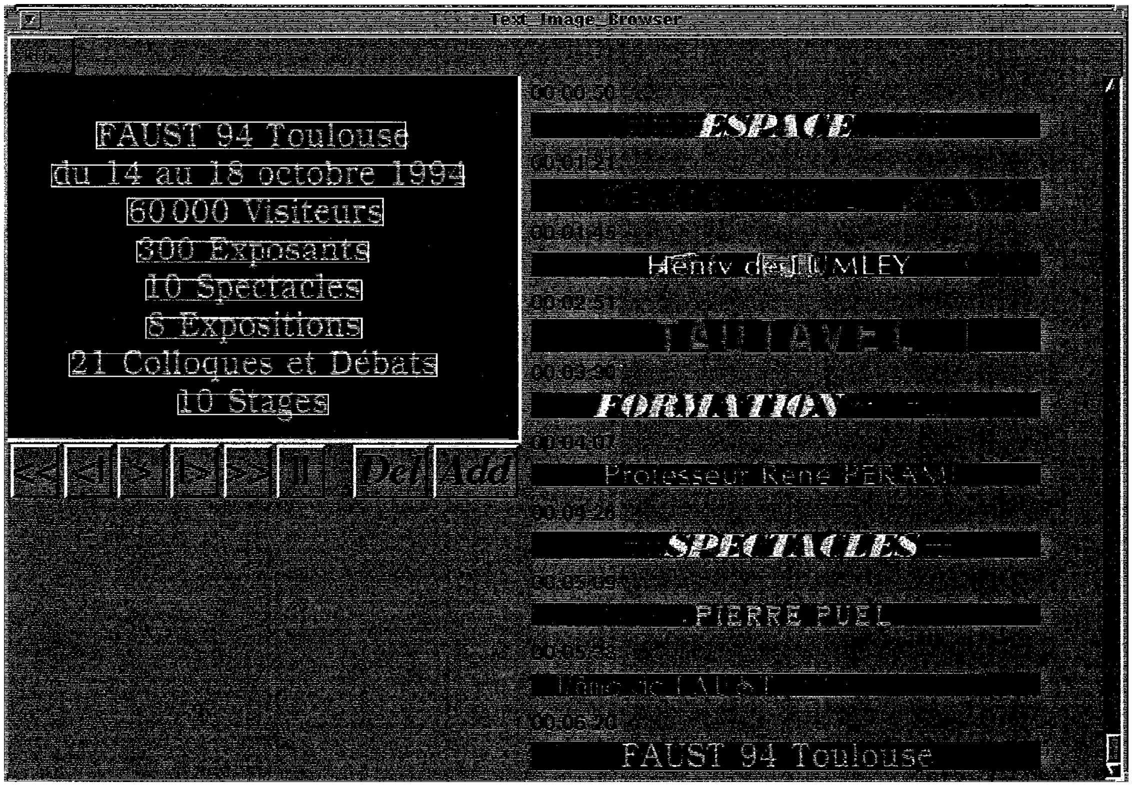

FIG. 3. Text image browser: text-image based video retrieval user interface (the source images are extracted from a video FAUST '94, directed by Maryse Baute and Veronique Dardoize, © Mairie de Toulouse, Jan. 1995).

The menu ''Film'' is used to select and load a video document with its text-line database obtained from the text detection algorithm. Video is displayed in the left video window and is manipulated by the conventional operating buttons of ''fast backward,'' ''step backward,'' ''play,'' ''step forward,'' ''fast forward,'' and ''stop.'' One representative text-image and the time code for each frame containing text-images are displayed in the right text-image scrolled window. The user can browse the document by scrolling the text-images in the scrolled window and get direct access in a video document to the temporal position of an interesting text-image selected by double clicking of the mouse. On the video window, if the frame contains text-lines, they are marked by enclosing boxes. A representative text-image can be replaced with another text-line by double clicking inside of the enclosing box of the replacing text-line in the video window. Text-lines can be eliminated from the database by clicking the ''Delete'' button after selecting enclosing boxes in the video window or text-images in the text-image scrolled window. A text-line which the detection algorithm has not detected can be inserted into the database by drawing an enclosing box around the text-line and clicking the button ''Add.'' The video window and the text-image scrolled window are updated immediately with the change of the text-line database. The human intervening cost for inserting a text-line which requires careful location of the text-line and drawing of the enclosing box is much more expensive than for deleting a text-line which does not need so much work. False negative detection is consequently more severe than false positive detection.

There are other applications of the text detection in video images. Larger scale segmentation than the shot using image and sound analysis have been proposed for more abstracted description, representation, and retrieval of video documents such as macrosegmentation by rules derived from the observation of film editing, based on local clues such as transition effects, montage rhythm and shot duration (Aigrain *et al.*, 1995), segmentation by *a priori* model of temporal and spatial structure of a specific news program using knowledge base (Zhang *et al.*, 1995), story unit extracting by finding cut edges from a scene transition graph in which nodes are groups of similar shots and each directed edge between two nodes represents the temporal relationship (Yeung *et al.*, 1995; Yeung *et al.*, 1996), and film genre detection by modeling image and sound characteristics of each genre (Ficher *et al.*, 1995). Detected text images can be used as useful clues for film genre detection of commercial sections, news programs, the generic sections of films by modeling size, position, frequency, and composition of text images for each genre, and with an OCR technology applied to recognize detected text-images, linguistic retrieval technologies such as keyword or natural language retrieval could be possible.

TABLE 1
Processing Time of the Color Plane Method

Operation	Time (s)
Color segmentation	2.2
Negative component elimination	4.8
Find text-boxes	0.6
Total	7.6

V. EXPERIMENTATION RESULTS

The text detection algorithm is implemented on Sun Sparc II workstation with a Parallax X-Video board connected. The resolution of the image was 384 x 288 with R, G, B channels of 256 levels for each. The heuristic parameters of the algorithm were set to the values explained in the Section III and the experimentation was performed on 50 images which contain 124 text-lines with various character styles and sizes with different backgrounds from various other video documents. Text-line detection rate was about 86% (107 text-lines detected) and weak color contrast or too-small characters were the main cause of the failure of the algorithm. Tens of thousands of colors in source images were reduced to within 7 colors after the color segmentation step. The spatial variance method is also implemented and tested for the same set of images for comparison (other existing text detection methods can not comply with the requirements for text detection of video documents due to binary segmentation, connected component analysis or OCR adopted for other kind of applications). Tables 1, and 2 show, respectively, the average computing time of the proposed color-plane method and the spatial variance method. The color plane method is observed faster than the spatial variance method.

VI. CONCLUSION AND FUTURE WORK

A text detection algorithm for video documents, a text-image-based video browser, and discussion on applications are presented. Input images are selected at fixed time intervals from shots detected by a scene-change detection method and are segmented by color clustering method around dominant colors. Global negative shape compo-

TABLE 2
Processing Time of the Spatial Variance Method

Operation	Time (s)
Spatial variance image	5.2
Canny edge detection	5.9
Find text-boxes	0.6
Total	11.7

nents are eliminated. For each color plane, text-lines are detected through candidate text-line detection, text-segment detection, and post-processing steps, and finally, the geometric and temporal position and the text-image of detected text-lines are stored in a database. With the text-image browser, the user can browse through text-images to get direct access in a video document to the temporal position of an interesting text-image and correct text-lines falsely identified by the detection algorithm. Experimental results show the algorithm effectiveness as a text detection method for indexing and structuring video documents. As the future work, automatic detection of commercial sections of TV program and of the generic section of film and extraction of statistics text-images from sports programs are planned as applications of the text detection algorithm for video documents.

REFERENCES

M. Agosti, M. Melucci, and F. Crestani, Automatic authoring and construction of hypermedia for information retrieval, Special Issue on Content-Based Retrieval, *Multimedia Systems* **3,** 1995, 15–24.

P. Aigrain and P. Joly, The automatic real-time analysis of film editing and transition effects and its applications, *Comput. Graphics.* **18,** 1994, 225–234.

P. Aigrain, P. Joly, and V. Longueville, Medium knowledge-based macro-segmentation of video into sequences, in *Proc. IJCAI Workshop on Intelligent Multimedia Information Retrieval* (Mark Maybury, Ed.), Montrèal, 1995.

T. Blum, D. Keislar, J. Wheaton, and E. Wold, Audio databases with content-based retrieval, in *IJCAI 95, Intelligent Multimedia Information Retrieval, 1995,* pp. 71–92.

A. S. Gordon and E. A. Domeshek, Conceptual indexing for video retrieval, in *IJCAI 95, Intelligent Multimedia Information Retrieval, 1995,* pp. 23–38.

S. Ficher, R. Lienhart, and W. Effelsberg, Automatic recognition of film genres, in *Multimedia '95, San Francisco, 1995,* pp. 295–304, 367–368.

L. A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics image, *IEEE Trans. PAMI* **10,** 1988, 910–918.

A. G. Hauptmann and M. A. Smith, Text, speech, and vision for video segmentation: The informedia project, in *IJCAI 95, Intelligent Multimedia Information Retrieval, 1995,* pp. 17–22.

P. Joly and H. K. Kim, Efficient automatic analysis of camera work and microsegmentation of video using spatio-temporal images, *Signal Process. Image Commun.* **8,** 1996.

S. S. Kuo and O. E. Agazzi, Keyword spoting in poorly printed documents using pseudo 2D HMMs, *IEEE Trans. PAMI* **16,** 1994, 842–847.

C. P. Lai and R. Kasturi, Detection of dimension sets in engineering drawings, *IEEE Trans. PAMI* **16,** 1994, 848–855.

Y. Lu, Machine printed character segmentation—An overview, *Pattern Recognition* **28,** 1995, 67–80.

A. Nagasaka and Y. Tanaka, Automatic video indexing and full-video search for object appearances, in *Proc. of the IFIP, Second Working conference on Visual Database Systems, Budapest, 1991,* pp. 113–127.

J. Ohya, A. Shio and S. Akamatsu, Recognizing characters in scene images, *IEEE Trans. PAMI* **16,** 1994, 214–220.

N. R. Pal and S. K. Pal, A review on image segmentation techniques, *Pattern Recognition,* **26,** 1993, 1277–1294.

R. W. Picard and T. P. Minka, Vision texture for annotation, Special Issue on Content-Based Retrieval, *Multimedia Systems* **3,** pp. 3–14, 1995.

J. Rocha and T. Pavlidis, Character recognition without segmentation, *IEEE Trans. PAMI* **17,** 1995, 903–909.

H. Sakamoto, H. Suzuki, and A. Uemori, Flexible montage retreival for image data, in *Proc. Storage and Retrieval for Image and Video Databases II,* SPIE, Vol. 2185, pp. 25–33, Int. Soc. Opt. Eng., Bellingham, WA, 1994.

P. Schauble and M. Wechsier, First experiences with a system for content based retrieval of information from speech, in *IJCAI 95, Intelligent Multimedia Information Retrieval, 1995,* pp. 59–70.

S. Sclaroff and A. Pentland, Modal matching for correspondence and recognition, *IEEE Trans. Pattern Anal. Machine Intelligence,* **17,** 1995, 545–561.

B. Shahraray and D. C. Gibbon, Automatic generation of pictorial transcripts of video programs, in *Multimedia Computing and Networking 1995,* SPIE, Vol. 2417. Int. Soc. Opt. Eng., Bellingham, WA, 1995.

T. Taxt, P. J. Flynn and A. K. Jain, Segmentation of document images, *IEEE Trans. PAMI,* **11,** 1989, 1322–1329.

D. Tegolo, Shape analysis for image retrieval, in *Proc. Storage and Retrieval for Image and Video Databases II,* SPIE, Vol. 2185, pp. 59–69, Int. Soc. Opt. Eng., Bellingham, WA, 1994.

T. Uchiyama and M. A. Arbib, Color image segmentation using competitive learning, *IEEE Trans. PAMI,* **16,** 1994, 1197–1206.

W. Wolf, Key frame selection by motion analysis, in *ICASSP '95.*

M. M. Yeung and B. Liu, Efficient matching and clustering of video shots, in *International Conference on Image Processing, 1995.*

M. M. Yeung, B-L Yeo, W. Wolf, and B. Liu, Video browsing using clustering and scene transitions on compressed sequences, in SPIE, Vol. 2417, pp. 399–413, Int. Soc. Opt. Eng., Bellingham, WA, 1995.

M. M. Yeung, B-L Yeo, and B. Liu, Extracting stroy units from long programs for video browsing and navigation, in *International Conference on Multimedia Computing and Systems, 1996.*

H. Zhang, A. Kankanhalli, and S. W. Smoliar, Automatic partitioning of full-motion video, *Multimedia Systems* **1,** 1993, 10–28.

H. Zhang, S. Y. Tan, S. W. Smoliar, and G. Yihong, Automatic parsing and indexing of news video, *Multimedia Systems,* **2,** 1995, 256–266.

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

HAE-KWANG KIM received his B.S. in electronic engineering from Hanyang University, Korea, in 1986 and worked for developing CD-ROM and WORM drives and CD-ROM multimedia applications at Samsung Electronics from 1986 to 1992. He received his DEA diploma in computer science from Paul-Sabatier University, France, in 1994, and is currently a Ph.D. candidate in computer science at Paul-Sabatier University, working as a student researcher at AMI lab., IRIT. His current research interests include video document processing, man–machine interface, pattern recognition, video compression, video library, and video editing tools.